

5-9-2014

# Crime in Game Theoretic Models: An exploration of the rational criminal in a variety of frameworks

Benjamin A. Chalmers  
*Wofford College*

Follow this and additional works at: <http://digitalcommons.wofford.edu/studentpubs>



Part of the [Behavioral Economics Commons](#), and the [Criminology Commons](#)

---

## Recommended Citation

Chalmers, Benjamin A., "Crime in Game Theoretic Models: An exploration of the rational criminal in a variety of frameworks" (2014). *Student Scholarship*. Paper 2.  
<http://digitalcommons.wofford.edu/studentpubs/2>

This Honors Thesis is brought to you for free and open access by Digital Commons @ Wofford. It has been accepted for inclusion in Student Scholarship by an authorized administrator of Digital Commons @ Wofford. For more information, please contact [stonerp@wofford.edu](mailto:stonerp@wofford.edu).

# Crime in Game Theoretic Models:

An exploration of the rational criminal in a variety of frameworks

Ben Chalmers

Senior Honors Thesis

May 9, 2014

**Table of Contents:**

Section 1: Introduction

Section 2: Deterrence and the Effect of False Arrests

2.1: Background on Kleiman and Kilmer's Model

2.2: Reproducing previous results

2.3: Incorporation of False Arrests

2.4: Conclusions

Section 3: Coordination games and choreographers

3.1 General coordination games

3.2 Coordination games with a choreographer

3.3 A simple model of a coordination game

3.4 Incorporating a choreographer into the model

3.5: Conclusions about the model

Section 4: General Conclusions and Future Work

## **Section 1: Introduction**

There is as much contention over the cause of crime as there is about how to solve it, and the two issues are inextricable from one another. While the idea of studying such a deeply social and humanistic issue through the ‘cold lens’ of mathematics may seem unorthodox or even unproductive to the layperson, the practice has become commonplace since Gary Becker’s introduction of the ‘rational criminal’ model in his paper *Crime and Punishment: An Economic Approach* in 1968<sup>1</sup>. The rational criminal model is a method of explaining the actions of a criminal not by attributing them to an inherent mental difference or deficiency, nor by some product of social repression, but as decisions to be calculated in the same manner as all economic choices: a rational consideration of the costs and benefits of committing such an action. That this seems a natural way of looking at crime in the present day is a testament to its effectiveness in understanding some aspects of crime, as it was by no means orthodox when it was first proposed; the dominant view prior to Becker’s contribution was that criminals were produced by a combination of mental deficiency/illness and social oppression<sup>2</sup>.

Since this model entered the mainstream, the field of economics has subsumed many areas of crime research, and indeed a variety of other societal phenomena which would have previously been claimed by sociology or psychology. This paper will focus on the prevention aspect: What methods are most effective in preventing or reducing crime, and what approaches have we yet to study? The methodology of the paper is as follows: First, a simple game is introduced in which rational actors use Bayesian Inference to decide whether or not to commit a crime. This game is then modified by adding a false arrest component. Secondly, a simulation of a pure coordination game is constructed, in order to determine the effectiveness of an unrelated choreographer at producing and maintaining certain equilibria.

## **Section 2: Deterrence and the Effect of False Arrests**

### **2.1: Background on Kleiman and Kilmer's Model**

In their paper titled "The dynamics of deterrence", Mark Kleiman and Beau Kilmer cite a variety of evidence which suggests that crime is self-reinforcing: Living in areas with high crime makes an individual more likely to commit a crime him/herself. This is not a particularly controversial assertion, but they extend this idea and suggest that this means that there are multiple equilibria when it comes to crime levels in a certain area. In other words crime will always exist at some nonzero, 'natural', level, but this level can be changed. They use this idea to suggest that current law-enforcement tactics are flawed at best and hopeless at worst, and suggest an alternative solution which they call 'dynamic deterrence'. Instead of dispersing all the resources of a police force over its entire jurisdiction at a relatively even level, the authors state that a more effective approach would be to focus the bulk of the police force's resources on one region in particular (presumably that with the highest crime rate) until the region's crime levels have dropped to a new, lower equilibrium. Once this has occurred, the force can shift its presence to the second-most crime-ridden area, and the process continues until crime is at a new lower level all over. An important aspect of this dynamic model is that regions are ranked by the police force, and this ranking is made public. If any of the areas which have been previously altered by the police force begin to revert to their previous high-crime equilibrium, the police prioritize this region until crime is again suppressed. Thus the ranking of regions does not change, and the police force return to a higher-ranked region as soon as its crime levels begin to rise.

The simplest model within Kleiman and Kilmer's paper is a repeated game involving one potential criminal (the 'rational actor') and a police force. The most basic assumption of the

rational criminal model- that committing a crime is the result of a calculation of the expected benefit vs. expected cost of doing so- is key to this simple game. There is a 'true cost' of committing the crime: the cost of punishment multiplied by the probability it will occur. Committing a crime does not guarantee that punishment will be received, as the police force has limitations. Kleiman and Kilmer set the cost at 300 and the true probability of punishment at 0.5, so that the true expected cost is  $150^3$ . (The authors chose a range of 0-300 in multiples of 30 for both the expected cost and the benefit within this model.) Both values are objective and unchanging, but the true probability is unknown to the actor. The actor has a prior belief about the probability of punishment, which the authors set at the correct value of  $p = 0.5$ , but uses Bayesian updating to modify this belief. Bayesian updating is the application of Bayes' Theorem, an important concept in probability theory which allows us to estimate the *true* probability of something, given what we observe. Its formal expression is  $p(A|B) = p(B|A)*p(A) / p(B)$ , meaning the probability of A *given* B is the same as the probability of B *given* A, multiplied by the probability of A over the probability of B. It allows us to gain insight into conditional probability- the probability that something happens *given* that something else has happened. In this context, the actor is estimating the probability of arrest when a crime is committed, given that individual's experience in the past.

The only portion of this game which is not deterministic is the 'compliance cost'- the cost to the actor of *not* committing this crime. This could also be stated as the benefit of committing the crime; the two are identical in the economic sense, due to the idea of an opportunity cost. An opportunity cost is the loss of the benefit you would have gained from doing something, so the opportunity cost of not picking up a \$20 bill on the street is exactly \$20. For each iteration of this repeated game, the authors used a randomly generated compliance cost between 0 and 300 using

a Poisson distribution function, which emulates a normal distribution centered at 150. The function differs from a normal distribution in that it is non-smooth; it is only defined for multiples of 30 between 0 and 300. If the actor's expected cost (initially 150) of committing the crime exceeded the randomly generated compliance cost, no action was taken. If the compliance cost was higher than the expected cost, the actor would commit the crime and then, with a probability of 50%, receive punishment. The actor's belief about the probability of receiving punishment was then updated using the basic Bayesian equation. Thus the presence of punishment raised this 'subjective probability' and the absence lowered it. This process was repeated for ten iterations each simulation, with the saved output of each simulation being the number of crimes committed.

## **2.2: Reproducing previous results**

We begin by replicating this basic game, in order to eventually modify and expand it. The authors gave only a cursory explanation of their algorithm, but we will reconstruct it as faithfully as possible using the programming language of Python. To determine the expected sanction cost, we create a Bayesian updating function for the actor and multiply the probability it generates by 300 (the true cost if probability of sanction = 1) each time. The randomized compliance cost is more complex and allows for varying methodologies, but we selected a Poisson distribution function which generated data that mirrored the results within the paper.

Initial trials were of single games with far higher iterations than the 10 used by Kleiman and Kilmer, since the actor is 'honing in' on the true probability of receiving a sanction with a higher and higher sample size in its Bayesian function. However, this produced results with an unexpectedly high variance, and this variance was not reduced regardless of how many iterations the actor performed (trials were run with 10, 100, 1,000, and even 100,000). This was a clear

sign that the process was in fact path dependent: results in the first few iterations had a stronger effect on the final result than subsequent ones, and this effect was not significantly mitigated by increasing the iterations. As a result, the method was modified to more closely adhere to traditional Monte Carlo simulations, defined as a high number of repetitions of an algorithm with probabilistic components. Iterations were kept at 10, and the average number of crimes committed was found for 100,000 repetitions of the game (This process was performed 10 times, to find 10 averages with a relatively small variance- this is the column titled “1 [No Change]” in Table 1 in the appendix). The successful replication of Kleiman and Kilmer’s basic game allowed for the subsequent modifications detailed in the next section.

### **2.3: Incorporation of False Arrests**

The decision to incorporate the probability of false arrest into the model was primarily due to an assertion in a paper by Schrag and Scotchmer<sup>4</sup> that false arrests are a major cause of the self-reinforcing nature of crime (this feedback element of crime is the driving force behind Kleiman and Kilmer’s suggested ‘dynamic deterrence’ policy). They argue that individuals who live in high-crime areas have a much higher likelihood of being arrested for crimes they did not commit, as everyone is statistically more likely to be a criminal in these areas. (This is a clear example of negative externalities being imposed by criminals on non-criminals, and the presence of externalities generally calls for government intervention.) The authors clearly state that this is a cause of the feedback nature of crime: “This phenomenon lies at the heart of the empirical literature of the 1970s that tried to understand how enforcement parameters such as the probability of arrest affect crime rates, given that crime rates also affect the probability of arrest”<sup>4</sup>. We initially intended to add this component to the model by incorporating it into the Bayesian updating function for the expected sanction cost, but realized that this was unrealistic:



Schrag and Scotchmer suggest that the subjective probability of being arrested for a crime one did not commit is less a function of prior experience (which would use Bayesian updating) and instead a result of the *type* of crime the person did not commit. They use the example of embezzlement by an executive: the probability of a false arrest is far lower than for that of a mugging or burglary, since the number of individuals able to commit the crime is far lower. There is also a more general problem that arises when trying to incorporate this into a Bayesian function: The function estimates the probability of actor  $x$  being arrested, *given* that actor  $x$  has committed a crime, but has no way of incorporating something outside of this conditional probability (the probability of arrest given that actor  $x$  has *not* committed a crime). Because the probability of false arrest seems to be based almost entirely off of situational circumstances rather than prior experience, we decided against modifying the expected sanction portion of the function and instead incorporated the possibility of false arrest into the compliance cost portion.

The probability of false arrest was implemented as a percentage of the compliance cost rather than a simple linear addition to it; it makes sense that crimes with higher compliance costs would also have a higher probability of false arrest, as these are crimes of opportunity and therefore appeal to a higher number of individuals. We found that a 1.0% increase in the compliance cost generated an increase of 18%-20% in the number of crimes committed, a result which was surprising and exciting (Data shown in Table 1 in appendix). We decided to test even smaller increases in the compliance cost, and found that even an increase by 0.01% generated an 18%-20% increase in the compliance cost as well, which was initially exciting: Did this result imply that the presence of the possibility of false arrest, no matter how small, increases an individual's likelihood of committing a crime? We quickly realized that this result was merely a result of the limitations of the simulation: Because the Poisson distribution function generates

integral amounts (in this case, each compliance cost was a multiple of 30), and initial expected sanction costs are integral as well (although they can eventually include fractions, as the numerator and denominator of the Bayesian fraction grow and create a longer decimal expansion), any discrepancy, no matter how small, removes the possibility that the expected sanction cost will equal the compliance cost. Because our model caused the actor not to commit a crime when the costs are equal (Kleiman and Kilmer didn't mention their choice for this scenario, but this is a reasonable assumption), any minor shift in the compliance cost above 0% is essentially the same effect as if we changed the actor's decisions so that he/she does commit a crime when the costs are equal.

## **2.4: Conclusions**

In light of these issues, a successful implementation of a false arrest component into this model would require changing the Poisson distribution function in favor of a continuous function, presumably the normal distribution. Because this model was extended to generate the main results of Kleiman and Kilmer's paper, any significant modification of it would require a deviation from their argument. Though the modifications to Kleiman and Kilmer's model provided little in the way of conclusive statements about criminal deterrence, they served as an introduction to game theoretic modeling. A desire to adhere to game theoretic model led to the notion of viewing crime as a coordination game, with the possible inclusion of an external choreographer.<sup>10</sup>

## **Section 3: Coordination games and choreographers**

### **3.1 General coordination games**

A Nash Equilibrium is the most important solution in Game Theory; it is a situation in a

game where no individual actor can improve his/her payoff by changing strategies, assuming he/she has the correct beliefs about the strategies used by other players. A pure strategy Nash Equilibrium is the simplest form of this, where each actor is playing one strategy as opposed to randomizing between two or more. A coordination game occurs when there are multiple pure strategy Nash Equilibria and both (or all) parties mutually benefit if 'coordination' can be achieved. A common example of this is the adoption of technological standards: which side of the road drivers use, which type of measurement system is used (standard vs. metric), and which format people use to watch home videos (Betamax vs. VHS, or more recently Blu-Ray vs. HD-DVD) are all most beneficial when the population agrees on which will be used- even if both formats are of equal inherent quality.

Crime can be described as a coordination game, to an extent: keeping all else constant, criminals face a lower cost of committing a crime if others are committing it as well, since their individual probability of arrest is lowered as a result. This adds weight to Schrag and Scotchmer's assertion that crime can have feedback qualities due to high-crime areas having higher probabilities of false arrest, raising the compliance cost for all citizens who live in them and increasing their likelihood of committing crimes themselves. In fact, even Kleiman and Kilmer's paper is based on the belief that crime begets more crime, and that criminals are better off if there are more criminals surrounding them. Viewing crime in this manner is by no means my idea; it was suggested by Dr. Pech and it was not long before I found a paper by Peter-J. Jost titled "Crime, coordination, and punishment: An economic analysis" which creates a model of a community, with the conclusion that "In this situation, the expected payoff of an individual from committing the offense is higher the more individuals also decide to behave illegally."<sup>5,10</sup>

### 3.2 Coordination games with a choreographer

Once we have accepted the idea that crime can be viewed as a coordination game of sorts, an extension of the idea becomes possible: The inclusion of an external choreographer. In a standard coordination game, actors have no information about the decisions made by the other actors (an equilibrium is easily achieved if signaling is allowed). The presence of a choreographer can change this, by functioning as a signal to all parties. A standard example is that of a stop light at an intersection: It resolves the coordination problem of drivers who want to cross an intersection at alternating times, but do not know when the other driver(s) intend to cross. For a choreographer to be obeyed, it does not have to be perfect; It just has to be more reliable than a complete absence of information. Solutions achieved by the choreographer are called “Correlated Equilibria”, and contain all of the Nash Equilibria which already existed but also (sometimes) new ones. A connection to the previous research arose in that the police force in Kleiman and Kilmer’s dynamic deterrence model functions as a choreographer of sorts, hoping to push individuals toward the low-crime coordinated equilibrium. (However, choreographers traditionally help determine which equilibrium is initially reached; the standard model of a pure coordination game does not imply that they can shift actors from one equilibrium to another. This is discussed in more depth later in the paper.) We began to seek other potential ‘choreographers’ of crime levels within a community, as the existence of a choreographer would lend further support to the notion that crime does indeed function as a coordination game.

There are several examples we can think of immediately: The previously mentioned police force, the absence of viable employment options, even the levels of personal interaction within a community could all theoretically shift crime from one equilibrium to another. These phenomena are not only well-studied within economics and a broad variety of other disciplines,

they are also far too complex and nuanced to be incorporated into a basic model. A desire to explore unorthodox choreographers led to an exploration into the possibility of weather as an external choreographer.

The effect of weather on crime has been studied, though not to the same extent as phenomena which can be influenced by policy. Nonetheless, there have been a variety of studies produced which show a correlation between crime rates and various environmental factors: from the obvious ones like temperature<sup>6</sup> and darkness<sup>7</sup>, to more obscure measurements like wind, humidity, and even barometric pressure<sup>8</sup>. It is worth noting that any correlation in these categories adds support to the notion of the rational criminal: a reduction in the quality of atmospheric conditions raises the cost of going outside and committing crimes, and the idea that criminals do in fact respond to this (even if the response is weak) is indicative of the cost-benefit analysis which occurs.

While it is clear why studying the effects of something beyond humanity's control may be less practical than something we can alter and influence, the complete separation between weather and the system it influences (crime) can be a blessing in disguise. The problem of bilateral relationships (A affects B, but B affects A) is always a concern when making statistical inferences, but the field of economics is especially fraught with relationships of this kind, due to the fact that its subjects (humans) are complex and inconsistent. Though weather may affect crime, crime of course has no effect on weather. Thus, despite the fact that a knowledge of weather's effects on crime would not likely lead to any significant changes in policy or behavior, its pure unilaterality is intriguing. However, this unilaterality could also be used to contradict the idea that weather functions as a choreographer: It could be asserted that the presence of a heat wave or thunderstorm has effects on crime not because it functions as a signal to those on the

verge of committing one, but because of its explicit effect on human behavior. It may merely be the case that a snowstorm reduces criminal activity for the same obvious reason it reduces all other outdoor activities: People spend less time outside when they are less comfortable. In this case, weather coordinates criminals (or actors performing other activities) not by signaling to them, but by altering the cost or benefit of performing a certain action. This means that it would not fit the standard model of a coordination game with a choreographer.

In the face of this modeling problem, the realization occurred that a choreographer can exist and be effective, even if it is entirely unrelated to the system it is helping to coordinate. We rule out weather here because we argue that it does directly affect the system, by altering the cost-benefit function of the actors involved. A traffic light also does not fit this criterion because even if its signals are pre-programmed and unaffected by traffic, it is related to the system in that its explicit purpose is common knowledge to all involved. For a pure coordination game with a multitude of strategies and actors, the individual actor's best chance is a shot in the dark: He/she/it must hope that all others make the same choice, without any knowledge of their preferences or likelihood of doing so. Thus anything that provides even the slightest 'signal' can provide a benefit if the actors begin to notice its presence over time. We can use the stock market as an imperfect example of this. If some traders have a belief about some signal- let's say the presence of the full moon, for example- this may result in coordination despite the fact that it has no 'true' effect on the values of shares. If enough traders believe that stock X will go up in value the morning after a full moon, this turns into a self-fulfilling prophecy where demand for the stock goes up and in turn increases its value. This is imperfect for a multitude of reasons: The first is that the stock market is not by any means a pure coordination game, although it does have some similar traits; the fact that stocks do have an inherent value and are not merely valued

based on popularity means that the framework is more complex. This fact also leads to the second shortcoming of this example, in that the assumption that this inflated price would stay at its new level and not be corrected by the market is both bold and likely untrue. Despite not having a clear example from the real world to model, we decided to create a generalized simulation of such a system to see if it functions as predicted.

### **3.3 A simple model of a coordination game**

The most standard coordination game, consisting of two actors and two strategies, seemed too simplistic for this scenario. The actors already have a fairly high likelihood of coordinating per trial and a weak external signal would either dominate or have no effect. (A basic example of a coordination game is a situation where two actors each flip a coin, and receive money only when both coins land on the same face. Their probability of success is 0.5, which is the same in this case once we understand that 'heads' and 'tails' are the two strategy options.) We also wanted to have a repeated aspect of the game, to allow for 'learning' of actors—a trend toward selecting a certain strategy when the signal is present. We started with 10 actors (actor 0-9) and 3 strategies (strategy A-C), as this seemed complex enough to allow for an effective signal but simple enough for an accessible model and code. Each actor had an initial strategy randomly selected between A, B and C with equal probability for each. Each actor's payoff for the first iteration was equal to the number of actors who selected the same strategy: For five actors who picked A, each received a payoff of 5; for two who picked B, each receive a payoff of 2, and so on. This allowed for the coordination aspect of the game, as the Pareto optimal outcomes occur when all actors play the same strategy, regardless of which strategy it is. (Pareto optimal means that there is no solution which improves the situation of one actor without

worsening that of another, in other words there are no other solutions which would be better for every actor involved.) We also modified the traditional model by stipulating that in each iteration, each actor is only aware of the payoff he/she receives, and has no idea about the other two payoffs (except that their sum will equal 10 minus the payoff the actor received). If an actor's payoff for strategy  $x$  was 5 or higher for the first iteration, he/she has no incentive to deviate as the other two strategies could not possibly provide a higher payoff. (Note: If 5 actors picked A and 5 picked B, one might argue that an actor who switches from A to B will be better off as the payoff for B will now be 6, but this isn't possible: the best the actor can do is randomize between strategies B and C, which yields an expected payoff of  $6/2$  and is inferior.) Likewise, an actor playing strategy  $x$  and receiving a payoff less than or equal to 3 has an incentive to deviate and randomize between the other two strategies, as the expected payoff of switching would be 4 due to the fact that the payoff for  $x$  would be 2 and that the other two payoffs add up to 8. The question then is what to do if the payoff received for strategy  $x$  is exactly 4? By following the same logic, the actor has no incentive to deviate. By changing strategies, the payoff for  $x$  is changed to 3 and the payoffs for the other two values sum to 7, resulting in an expected payoff of 3.5 for deviation.

The framework of the game was set by the following system: actors receiving a payoff  $\geq 4$  in iteration  $n$  did not change strategies for iteration  $n + 1$ , actors receiving a payoff  $\leq 3$  randomized between the other two strategies for the subsequent iteration. The game ran for eight iterations and functioned as intended: the final outcome was a Nash Equilibrium. The most common result was for all actors to pick the same strategy and receive a payoff of 10. However, there were also outcomes where 5 players played one strategy and 5 played another, or 4 played one and 6 played the other. These would not conventionally be Nash Equilibria (not even the 5



and 5 solution, since switching from one to the other improves your payoff to 6- though it could be argued that this is an unstable mixed strategy Nash Equilibrium since all actors will have incentive to switch, resulting in another 5 and 5 split), but the stipulation that each actor cannot see the payoff for the other two outcomes causes them to be stable due to the possibility of receiving a 0 payoff. We then altered the threshold for changing strategies and modified the code so that players with a payoff  $\leq 4$  would switch, and those with a payoff of 5 or greater would not. This eliminated the 4 and 6 split, but also interestingly reduced the probability of a split equilibrium altogether; almost all trials (19 out of 20) resulted in all actors selecting the same final strategy. This led to the realization that any deviation threshold higher than 5 would have to result in a uniform final strategy, it would just take more iterations on average to arrive at this result. The average number of iterations necessary to reach this stable result increased exponentially with the deviation threshold: for a threshold of 8, only approximately half of the tests reached the equilibrium after 1,000 iterations. For a threshold of 9, this was the case at 10,000 iterations (all 40 trials for a threshold of 8 resulted in the stable equilibrium for this number of iterations).

### **3.4 Incorporating a choreographer into the model**

There are a variety of ways in which a choreographer component could be added to this simulation, and the trial and error process of doing so actually led to some insight about the efficacy of a choreographer in a pure coordination game. There were only two stipulations we made concrete for the choreographer: The first is that its signal had to be random, because the choreographer is supposed to be completely unaffected by the game. We decided on a straightforward binary on/off signal, in other words the choreographer 'signaled' by being on

(equal to 1) and had no effect when off (equal to 0). The second stipulation is that only a fraction of the actors had to have the belief that the choreographer had any effect: If a majority of the actors believed that the presence of the signal indicated that strategy  $x$  is preferable, the game would be over as soon as the signal first appeared, and the results would be predictable and uninteresting.

It was immediately clear that the game in the original coordination model was too limited to demonstrate any effect from a choreographer. With only three possible strategies, an equilibrium was generally reached within 10 iterations of the game when the threshold for 'sticking with' a strategy was a payoff of 5. Our first impulse was to make the most simple modification to the game, and increase this payoff threshold to 6, 7, or 8 and then see if a randomly-firing choreographer helped speed up the rate at which an equilibrium was reached. This was inspired by the realization that increasing the payoff threshold resulted in an exponential increase on the number of iterations necessary to reach equilibrium, and we thought a choreographer could significantly reduce this number. The most obvious shortcoming of this modification is that it negates any previous attempts at making actors adhere to game theoretic principles of strategy- it makes no sense for an actor in a repeated game to switch when his/her payoff is  $> 5$ , as he/she is *guaranteed* a lower payoff.

This initial modification shed light on a more general problem: In a repeated pure coordination game with a significant number of actors, a choreographer is highly restricted in its ability to influence the eventual outcome once a trend towards a certain strategy has begun. If it confirms the equilibrium the actors are already approaching, it is redundant, and if it gives any alternate suggestion this will likely result in unproductive noise. To use an example from earlier, once VHS has taken a significant enough share of the market over Betamax, a signal to

consumers to buy Betamax will have little effect. Thus a choreographer's impact is only significant in the early stages of the game: It can guide actors toward an initial trend which leads to an equilibrium, but its signal turns to useless 'cheap talk' once a clear trend has been established. We had initially intended to incorporate an evolutionary aspect of the game, with actors who received higher payoffs in early iterations somehow passing this advantage on (to themselves or perhaps by influencing others). However, we quickly realized that adding a 'learning component' to the actors meant that they had little control over which equilibrium initially began to arise, and decided to abandon this approach as well. This had ramifications for how we approached the coding aspect of the problem, but was also helpful in allowing us to understand which situations would be affected by a choreographer and which would not.

With all of these discoveries in mind, the first modification we made to the original model was to increase the number of strategies from three to six, so that strategies A-F were now possible. We kept the number of actors at 10, so that equilibrium strategies would not start to appear as quickly as they would with more actors. The game still functioned more or less as before, with the payoff for each actor in each iteration being equivalent to the total number of actors playing the same strategy. Each actor was given a randomly selected initial strategy choice. The cutoff payoff remained at 4: Actors receiving more than this payoff did not change their strategy, actors receiving 4 or less did, by randomizing between the five strategies they had *not* selected in that iteration. (Note: We did perform trials with a cutoff of 5, with predictable results-fewer equilibria were reached in the same amount of iterations, but the presence of a choreographer's signal had the same effects. See Table 2 in the appendix for full results.) We ran trials with no choreographer to get a sense of the average iterations necessary to reach an equilibrium, and to ensure that each equilibrium (all play A, all play B, etc.) occurred with the

same frequency. Just like the last model, each game consisted of 50 iterations, and 1,000 games were run per simulation. The simulation recorded the total number of equilibria reached, as well as the total going toward strategy A (as this would be the strategy influenced by the choreographer).

The number of actors who took note of the choreographer's signal was originally set at three, and each actor was conditioned to play strategy A when the signal was present. Because we eliminated the learning component of this game, the signal either occurred at the beginning of the game or it did not; it did not fire randomly during the actual iterations. This actually meant that its effect was only significant in the first iteration, where its presence set the strategy for three of the actors to be A. This corresponds with the notion that a choreographer has influence at shifting individuals toward an equilibrium, but not at changing them from one to another. With a payoff threshold of 4, the presence of the choreographer would be expected to have a huge effect: Causing three actors to play strategy A means that only one of the remaining seven actors must randomly choose A in order for it to start trending towards an equilibrium strategy- this occurs with probability of approximately 0.79. The results confirmed this prediction: when the choreographer signaled before the first iteration, Strategy A occurred as an equilibrium almost as much as the other four strategies combined. Of the average 979.6 equilibria per 1,000 games, 432.4 were with all actors playing strategy A, or 44.14% of the equilibria (Table 2).

We wanted to test the efficacy of the choreographer's signal when it had less influence on the system, and adjusted the model in two ways: The first was by raising the payoff threshold from 4 to 5, and the second was by reducing the number of affected actors from three to one. Even with these alterations, the choreographer still had a clear effect on the simulation. The average number of equilibria per 1,000 games was virtually unchanged by the presence of a

signal: 448.0 without one, 448.2 with one. However, the number of equilibria with all actors choosing A increased from 73.1 to 80.2 per 1,000; a shift in percentage of total equilibria from 16.3% to 17.9%. While these results are less extreme, they are meant to indicate the weakest possible signal within the framework we allowed: A low payoff threshold, only one actor who responds to the signal, and a signal which is only present at the first iteration.

The result of this simulation- that a signal completely unrelated to the system can and does have an effect on which eventual equilibrium is most likely to occur- is in large part due to the complete randomness of every other component involved. In a pure coordination game, any factor which differentiates one equilibrium from the others is invaluable to the players involved. The economist Thomas Schelling introduced the concept of these 'differentiated equilibria' in the form of focal points in 1960<sup>9</sup>. He defined them as a solution which seems normal or natural to people, and would be so for others as well. In this model, the existence of a common understanding or belief is unnecessary, as the adherence of a single individual to a belief is enough to sway the equilibrium towards it.

### **3.5: Conclusions about the model**

Though a choreographer has little influence on the outcome once a coordination game has started trending toward a particular equilibrium, the extent to which it can affect the outcome in the very early stages is encouraging for those who wish to enact policy or guide the future in some way. While it may be difficult to spot a coordination game before it has already begun, there must exist a window of time in which individuals or groups can have a large influence on the eventual equilibrium. If we are to extend this reasoning to the early analysis of Kleiman and Kilmer's dynamic deterrence model, we could assert that the authors are pursuing the wrong end

of a coordination game: Instead of spending time and resources with the intention of shifting from one equilibrium to another, perhaps research could be done into regions or demographics where crime is suddenly becoming an issue. A proactive approach to such an area may well provide confirmation of their assessment that crime rates exist at multiple equilibria, as well as offering an example of the power of influencing coordination games so early.

#### **Section 4: General Conclusions and Future Work**

There is room for expansion and improvement within the coordination model we made. We found that the complete independence of the choreographer from the system meant that there were a variety of options regarding its incorporation to the model, and we would be interested in testing more of these. One shortcoming about the way it is currently included is that its function is virtually indistinguishable from an individual's preference; the idea that it signals randomly is more or less negated by the fact that it only has an effect in the first iteration, meaning the same results could be obtained by running side-by-side simulations of the game with and without the signal. We would be interested to see the effect of allowing the choreographer to signal during the iterative process, but had a hard time making this work. A functional version of this would allow for an objective analysis of the importance of choreographers acting early, as signals used in early iterations could be tested against ones occurring later on. Another area for expansion is to literally expand the model, by adding more actors and/or more strategies. This became laborious very quickly, and would likely require some creative coding to mirror large-scale synchronizations and activities.

It was also suggested<sup>10</sup> that crime as described in this paper functions less as a pure coordination game and more as a stag-hunt game, another ‘basic game’ in Game Theory that falls under the class of coordination games. The difference is that in a stag-hunt game, various equilibria are *not* of equal payoff to the actors involved. It makes sense that actors would be more likely to commit crime if others are doing the same and to avoid it if others are doing so, but the actor will likely receive a different payoff depending on which of these coordinations occur. In this case, the act of committing a crime is akin to the decision to hunt the stag, while the ‘safe’ option of forgoing a crime is akin to hunting the rabbit.

As for expansions outside of this specific model, the most meaningful contribution would be empirical results of some kind. To our knowledge, no police force has attempted Kleiman and Kilmer’s “Dynamic Deterrence” model on a large scale, but an econometric analysis of such a policy would be invaluable. There is also the potential to explore the relationship between weather and crime, specifically the potential role of weather as a choreographer, using econometric analysis. Such an approach could provide insights about the roles of external choreographers, as well as shedding light on the rational preferences of criminals.

**Appendix:**

**Table 1: Effect of False Arrests on crimes committed per 10 opportunities**

False Arrest Multiplier:	1 (No Change)	1.01	1.0001
Trial 1	4.13781	4.92591	4.92215
Trial 2	4.13402	4.9248	4.93282
Trial 3	4.12522	4.93111	4.92462
Trial 4	4.13648	4.92663	4.91483
Trial 5	4.1254	4.91865	4.92458
Trial 6	4.13182	4.93864	4.93849
Trial 7	4.12672	4.9286	4.91344
Trial 8	4.12437	4.92153	4.93618
Trial 9	4.13439	4.91717	4.9252
Trial 10	4.13073	4.92551	4.92048
Average:	4.130696	4.92855	4.925279
Variance:	$2.488*10^{-5}$	$3.850*10^{-5}$	$7.037*10^{-5}$



**Table 2: Effect of a Choreographer on simulation, with threshold of 4 or 5 and one or three actors responding to its signal.**

	Threshold 4, no choreographer		Threshold 4, Choreographer present and one actor responding		Threshold 4, choreographer present and three actors responding		Threshold 5, Choreographer present and three actors responding	
	Equilibria Reached/ 1,000 trials	Equilibria with Strategy A	Equilibria Reached/ 1,000 trials	Equilibria with Strategy A	Equilibria Reached/ 1,000 trials	Equilibria with Strategy A	Equilibria Reached/ 1,000 trials	Equilibria with Strategy A
Trial 1	971	156	972	183	986	415	501	166
Trial 2	977	165	975	203	981	432	502	165
Trial 3	973	156	975	192	988	438	511	158
Trial 4	982	154	979	186	978	412	488	152
Trial 5	977	171	973	190	973	434	482	159
Trial 6	977	180	978	190	975	457	493	171
Trial 7	978	155	973	191	983	445	499	157
Trial 8	976	150	975	160	984	442	510	147
Trial 9	973	167	975	207	970	437	502	160
Trial 10	976	160	979	207	978	412	505	165
Average	976	161.4 (16.5%)	975.4	190.9 (19.5%)	979.6	432.4 (44.1%)	499.3	160 (32.0%)

## Python Code: Deterrence Function

```
# File: deterrencemodel.py
```

```
import math
import random
```

```
def poisson_probability(actual, mean): # A function which creates a poisson distribution function
with k and lambda being the inputs.
```

```
# Ben's comment: This particular function (poisson_probability) is the only work that is not my
own.
```

```
# I found it on a comment board when I googled 'poisson distribution python'. I assume this
means it
```

```
# is 'common property' for programmers to use, but I am not sure of the specific ethics in coding,
# and I therefore feel the need to mention that this function is not my work. Everything after this
# function is all mine.
```

```
# naive: math.exp(-mean) * mean**actual / factorial(actual)
```

```
# iterative, to keep the components from getting too large or small:
```

```
p = math.exp(-mean)
for i in range(actual):
    p *= mean
    p /= i+1
return p
```

```
def random_pick(j): # A function which generates a number based on the probabilities
determined by the poisson distribution function.
```

```
k = 0
x = random.uniform(0, 1)
```

```
probkCounter = 0
```

```
for k in range(0,20):
    probk = poisson_probability(k,5)
```

```
    if probkCounter >= x:
        return k-1
```

```
    else:
        probkCounter += probk
```

```
    return k
```

```
def sanction_function():
```

```
    j = 0
```

```

sanctionCost = 300
# The following two values are the numerator and denominator of the Bayesian probability
function,
# which will be updated as the criminal commits crimes and updates his/her probability of
sanction
# upon observing whether or not he/she is sanctioned each time.
bayesNum = 1
bayesDenom = 2

for i in range(10):

    complianceCost = 30 * random_pick(j)

    subjectiveProb = bayesNum * (1/bayesDenom)
    expectedSanction = sanctionCost * subjectiveProb

    print("Trial", i+1)
    if complianceCost <= expectedSanction:

        print("The compliance cost is ", complianceCost , " and the expected sanction cost is ",
expectedSanction , " so the actor does not offend this period.")
        elif complianceCost > expectedSanction:

            print("The compliance cost is ", complianceCost , " and the expected sanction cost is ",
expectedSanction , " so the actor does offend this period.")
            sanction = random.choice([True, False]) #Note: the true sanction probability is always .5,
so this doesn't change.
            if sanction == True:
                print("The actor was sanctioned this period, and updates his/her subjective probability
of sanction accordingly.")
                bayesNum += 1
                bayesDenom += 1
                subjectiveProb = bayesNum * (1/bayesDenom)
            else:
                print("The actor was not sanctioned this period, and updates his/her subjective
probability of sanction accordingly.")
                #bayesNum does not change
                bayesDenom += 1

    else:
        print("error")

sanction_function()

```

## Python Code: Coordination Game

# File: CoordinationSync

```
import math
import random
```

```
class Actor(object):
    # Creates an 'actor' object which will be affected by the coordination game.
    stratNum = 0
    payoff = 0
```

```
def make_Actor(stratNum, payoff):
```

```
    actor = Actor()
    actor.stratNum = stratNum
    actor.payoff = payoff
    return actor
```

```
def ListUpdater(actorList):
```

```
# Performs one iteration: Takes actors, changes their payoffs to reflect number of other
# actors receiving the same payoff. Also returns a numerical value for how many actors
# played each strategy
```

```
    payoff_a, payoff_b, payoff_c, payoff_d, payoff_e, payoff_f = 0, 0, 0, 0, 0, 0
    List_a = []
    List_b = []
    List_c = []
    List_d = []
    List_e = []
    List_f = []
```

```
    for i in range(len(actorList)):
```

```
        if actorList[i].stratNum == 0:
            payoff_a += 1
            List_a.append(actorList[i])
```

```
        elif actorList[i].stratNum == 1:
            payoff_b += 1
            List_b.append(actorList[i])
```

```

elif actorList[i].stratNum == 2:
    payoff_c += 1
    List_c.append(actorList[i])

elif actorList[i].stratNum == 3:
    payoff_d += 1
    List_d.append(actorList[i])

elif actorList[i].stratNum == 4:
    payoff_e += 1
    List_e.append(actorList[i])

else:
    payoff_f += 1
    List_f.append(actorList[i])

for i in range(len(List_a)):
    List_a[i].payoff = payoff_a

for i in range(len(List_b)):
    List_b[i].payoff = payoff_b

for i in range(len(List_c)):
    List_c[i].payoff = payoff_c

for i in range(len(List_d)):
    List_d[i].payoff = payoff_d

for i in range(len(List_e)):
    List_e[i].payoff = payoff_e

for i in range(len(List_f)):
    List_f[i].payoff = payoff_f

return (List_a, List_b, List_c, List_d, List_e, List_f, actorList)

def StratUpdater(StratA, StratB, StratC, StratD, StratE, StratF):
# Actors who picked losing strategies randomize between the strategies they didn't pick

if len(StratA) <= 4:

```

```

    for i in range(len(StratA)):
        StratA[i].stratNum = random.choice([1,2,3,4,5])
    else:
        StratA == StratA

    if len(StratB) <= 4:
        for i in range(len(StratB)):
            StratB[i].stratNum = random.choice([0,2,3,4,5])
        else:
            StratB == StratB

    if len(StratC) <= 4:
        for i in range(len(StratC)):
            StratC[i].stratNum = random.choice([0,1,3,4,5])
        else:
            StratC == StratC

    if len(StratD) <= 4:
        for i in range(len(StratD)):
            StratD[i].stratNum = random.choice([0,1,2,4,5])
        else:
            StratD == StratD

    if len(StratE) <= 4:
        for i in range(len(StratE)):
            StratE[i].stratNum = random.choice([0,1,2,3,5])
        else:
            StratE == StratE

    if len(StratF) <= 4:
        for i in range(len(StratF)):
            StratF[i].stratNum = random.choice([0,1,2,3,4])
        else:
            StratF == StratF

    return (StratA + StratB + StratC+ StratD + StratE + StratF)

def main():

    choreoCounter = 0
    nonchoreoCounter = 0
    equilibriaReached = 0

```

```

stratAcounter = 0

for i in range(1000):
    actor1 = make_Actor(random.randrange(0,6), 0)
    actor2 = make_Actor(random.randrange(0,6), 0)
    actor3 = make_Actor(random.randrange(0,6), 0)
    actor4 = make_Actor(random.randrange(0,6), 0)
    actor5 = make_Actor(random.randrange(0,6), 0)
    actor6 = make_Actor(random.randrange(0,6), 0)
    actor7 = make_Actor(random.randrange(0,6), 0)
    actor8 = make_Actor(random.randrange(0,6), 0)
    actor9 = make_Actor(random.randrange(0,6), 0)
    actor0 = make_Actor(random.randrange(0,6), 0)

# This section is commented out because I removed element of randomness.
# choreographer = random.choice([0, 1]) #Decides whether choreographer signals
# if choreographer == 1:
#     choreoCounter += 1
#     actor1 = make_Actor(0,0)
#     actor2 = make_Actor(0,0)
#     actor3 = make_Actor(0,0)

# else:
#     nonchoreoCounter += 1

actorList = [actor1, actor2, actor3, actor4, actor5, actor6, actor7, actor8, actor9, actor0]
# Creates list of all actor objects, each with a strategy and payoff from playing that strategy

StratA, StratB, StratC, StratD, StratE, StratF, actorListNew = ListUpdater(actorList)

for i in range(50): #Runs 50 iterations of single game
    actorList2 = StratUpdater(StratA, StratB, StratC, StratD, StratE, StratF)
    StratA, StratB, StratC, StratD, StratE, StratF, actorListNew = ListUpdater(actorList2)

    if len(StratA) == 10 or len(StratB) == 10 or len(StratC) == 10 or len(StratD) == 10 or
len(StratE) == 10 or len(StratF) == 10:

        equilibriaReached += 1 #Declares if an equilibrium is reached

```

```
else:
    equilibriaReached == equilibriaReached

if len(StratA) == 10:
    stratAcounter += 1

    print("Equilibria Reached? ", equilibriaReached, "Total for Strategy A:", stratAcounter,
          "Choreographer fired: ", choreoCounter)
    # print(len(StratA), len(StratB), len(StratC), len(StratD), len(StratE), len(StratF), '\n')

# print(numStratA, numStratB, numStratC, actorListNew[3].payoff)

main()
```



**Works Cited:**

1. Becker, Gary S. "Crime and Punishment: An Economic Approach." *Journal of Political Economy* 76:2 (1968). p.169. Print.
2. Raine, Adrian. "The Criminal Mind." *The Wall Street Journal*. April 26, 2013. Accessed online: <http://online.wsj.com/article/SB10001424127887323335404578444682892520530.html>
3. Kleiman, Mark and Kilmer, Beau. "The Dynamics of Deterrence." *PNAS* 106:34 (2009). pp 14230-14235.
4. Schrag, Joel and Scotchmer, Suzanne. "The Self-Reinforcing Nature of Crime." *International Review of Law and Economics* 17:325-335, 1997.
5. Jost, Peter-J. "Crime, coordination, and punishment: An economic analysis." *International Review of Law and Economics* Volume 21, Issue 1, March 2001, pp. 23–46.
6. Cohn, Ellen G. and Rotten, James. "Weather, Seasonal Trends and Property Crimes in Minneapolis, 1987-1988. A Moderator-Variable Time-Series Analysis of Routine Activities." *Journal of Environmental Psychology* (2000) 20, pp. 257-272.
7. Tompson, Lisa and Bowers, Kate. "A Stab in the Dark?: A Research Note on Temporal Patterns of Street Robbery." *Journal of Research in Crime and Delinquency* 2013 50: 616.
8. Cohn, Ellen G. "Weather and Crime." *British Journal of Criminology* Vol. 30 No. I. Winter 1990.
9. Schelling, Thomas C. "The strategy of conflict" (First ed.). Cambridge: Harvard University Press. 1960.
10. Conversation with Dr. Wesley Pech.