Cross-Cultural Decision Making, Vol. 44, 2022, 36–43
https://doi.org/10.54941/ahfe1001853

Approaches to Extending Game-Theoretic Analyses to Complex, Real-World Scenarios

W. Scott Neal Reilly and Leonard Eusebi

Charles River Analytics, Cambridge, MA 02138, USA

ABSTRACT

We present a novel methodology to describe and solve formal games that change some standard assumptions of game theory to make it easier to describe, solve, and analyze real-world adversarial scenarios. We describe a software implementation of this methodology that helps analysts understand the various possible outcomes of these situations. Finally, we describe some initial evaluations we have undertaken to demonstrate the usefulness of this approach to better understanding and reasoning about real-world scenarios.

Keywords: Game theory, Multi-objective optimization, Gray zone conflicts, Decision support tools, Competition below armed conflict

INTRODUCTION

In domains ranging from military engagements to business to politics to games, competitors take actions to gain an advantage over others. Game theory has been used extensively since the middle of the 20th century (e.g., (von Neumann and Morganstern, 1944; Nash, 1950; Luce and Raiffa, 1957)) to analyze such domains and to gain insights into the best moves for all competitors. While it is a powerful tool for analysis, game theory often falls short when applied to real-world encounters. Game-theoretic approaches over-simplify by assuming each side is composed of rational actors that attempt to maximize a single-valued utility function. Even with that simplification, real-world scenarios are often difficult to formalize as a solvable "game." And for scenarios that can be defined as a game, it can be computationally expensive to calculate the best actions for each actor for complex games.

We present a novel methodology that changes some standard assumptions of game theory to make it easier to describe, solve, and analyze real-world adversarial scenarios. We describe a software implementation of this formalism that helps to understand the various possible outcomes of these scenarios. Finally, we describe some initial evaluations we have undertaken to demonstrate the usefulness of this approach to better understanding and handling real-world scenarios.

NOVEL VARIANT OF GAME THEORY

Part of the motivation for the work we describe in this article is to better analyze gray zone conflicts (also known as Competitions Below Armed Conflict). These are typically competitions between two or more nation states in a variety of domains short of armed conflict. For instance, the current competition between China and the United States is playing out on political, social, and economic fronts as both sides understand the significant costs to a militarized engagement. In this section, we describe some of the ways we have modified traditional game theory to better support analyses of such conflicts.

Multi-Objective, Rational Actors

It is common in analyzing these kinds of conflicts to believe that one side or the other is acting irrationally and so any analysis tool that assumes rational actors can provide minimal guidance at best. Herbert Simon (1955) argued that assuming fully rational action isn't useful, but proposed a concept of *bounded rationality* that recognizes that decision makers will often make decisions that appear to be irrational because they lack information or time, not rationality. We argue one step further that the appearance of irrationality can often be attributed to not fully understanding the multiple objectives against which other actors are weighing their decisions.

We have created a multi-objective variant of game theory to include multiple forms of utility for each actor. This enables us to recast traditional, albeit simple game-theory games like the Prisoners' Dilemma and the Ultimatum Game, which produce results at odds with real-world expectations when confined to traditional measures of utility (i.e., minimizing jail time and maximizing money). By adding utility measures like commitment and fairness, we can generate a Pareto-optimal set of solutions that are better at recreating and explaining real-world behavior than traditional single-utility game theory. In our formulation, the actors are still acting rationally, they are just factoring in a more complex set of tradeoffs that our multi-utility game theory can naturally model.

Modeling Real-World Action Interactions

Our game representation scheme enables scenario modelers to express real-world action-to-action constraints like "enables" and "blocks." This makes it possible to capture important interactions between actions, where the actions can be those of a single actor or of different actors. For instance, we can describe situations where one actor does something to block another actor or one actor must choose from a set of mutually-exclusive responses to an adversary's actions. As we will describe below, accounting for these constraints also significantly reduces the space of possible solutions, making it tractable to find exact solutions for certain classes of complex scenarios. Our representation includes the following constraints:

- **One is required**. Links two actions where at least one of the two must happen.

- **Mutually exclusive.** Only one of a group of actions can happen.
- **Blocks.** If the first action in the pair happens, the second cannot happen.
- **Must co-occur with**. For a pair of actions, either both actions happen or neither happens.
- **Enables.** The second action can only happen if the first does.
- **Forces.** If a first action is taken, the second must also be taken.
- **Forces if not.** If a first action is not taken, then the second action must be taken.
- **At least one.** For two or more actions, at least one must be taken.
- **Exactly one.** For two or more actions, exactly one must be taken.
- **All or none.** For two or more actions, either all of them are taken or none are.

Modeling Time Dependence

The constraints just described also support basic reasoning about issues like ordering of actions without having to build full search trees or reason about time generally. This significantly simplifies the human modeling and automated reasoning tasks while not reducing all games to one-shot interactions like the Prisoners' Dilemma. We have found this to be a powerful technique that lets us reason about time at a useful level of abstraction that works well for many types of real-world scenarios.

Modeling Probabilities

Our approach does not explicitly handle probabilities, though this was an intentional design choice. We use expected values of outcomes to implicitly factor in probabilities, and we can model "chance" or "the environment" as actors with actions that are factored into the analysis, creating outcome cases where the actions happen and do not happen, though the model does not consider how likely they are to happen. Our intention with this simplification is to encourage users to analyze any possibility that they think is likely enough to put into the model, even if it is not likely to happen. With this as the goal, adding explicit probabilities only complicates the modeling and analysis processes without, we believe, sufficient analytical benefit. Relatedly, instead of a probability distribution, we support the specification of min and max values for the impact of an action on various motivations. This requires an analysis of the full range of possible outcomes instead of just the most likely ones.

As an example, say we wanted to analyze the question: "Should I set my alarm early to have time to shovel in the morning or should I sleep late because it will probably be enough for a snow day?" We could either create a chance node and analyze the optimal choice when it is enough for a snow day and the optimal choice when it is not. Or we could use expected values to model the min-max range of expected costs/benefits based on the range of probabilities of there being a snow day multiplied by the costs/benefits of sleeping in or getting up early.

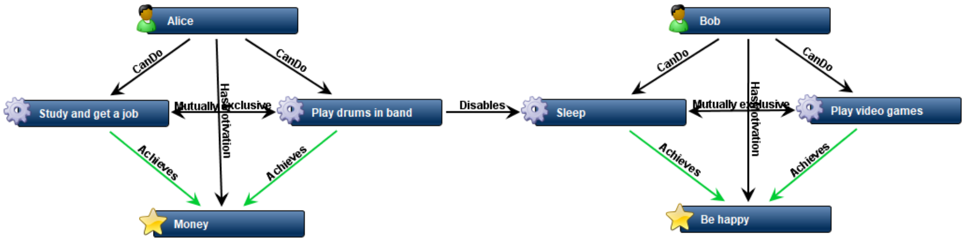

Figure 1: Simple scenario.

Identifying Only Pure Solutions

In game theory, a pure solution is one where there is a definitive choice of actions. A mixed solution is one where the actions are only probabilistic. Rock-Paper-Scissors (RPS) is a simple example where there is no pure solution, only a mixed solution with each of the actions being taken with 1/3 probability. Our software will find that there are no “winning” solutions to RPS. We have chosen to only identify pure solutions to the scenarios built with our tool for two reasons. First, this dramatically simplifies the solver and means we can find solutions to much larger, real-world scenarios. Second, in the gray zone conflicts we have been analyzing, the scenarios considered will play out in the real world only once, which means that a mixed solution would involve effectively rolling dice to choose the single course of action that is followed, an unpalatable approach when the stakes are high and actions may need to be defended after the fact. This has led us to believe that the downsides of identifying mixed solutions are greater than the benefits.

PUTTING IT ALL TOGETHER

Figure 1 provides a simple demonstration of what a scenario looks like in our tool. The analyst user creates a graph with nodes that represent the actors, the actions each actor can take, the motivations that each actor has, and the cost/benefit each action has for each relevant motivation. The links associate the actions and motivations with the actors, represent the action constraints described above, and associate the actions with the motivations they affect. We see in the figure that actions can impact the actions of other actors as well as actions of the self actor. Not shown in the figure is that actions can also impact the motivations of other actors. Also not shown is that the action-to-motivation links have values associated with them that indicate the value of the associated cost/benefit.

EFFICIENT & USABLE IMPLEMENTATION

Classical game theory assumes each combination of actions and each ordering of actions (in extensive-form games) has its own costs/benefits to each actor. This would result in

$$m \sum_{k=0}^{n} \frac{n!}{(n-k)!}$$

costs/benefit values to be specified where n = total number of actions, k = number of actions that might be taken in any given outcome, m = number of motivations. This is factorial growth in the size of the game specification and specifying a full O(m*n!) game is not practical from a usability perspective. Even if we could simplify the process of specifying an O(m*n!) game, it is still not computationally tractable to solve it.

As we noted, standard game theory assumes that each combination of actions and ordering of actions can have its own cost/benefit value (or, in our case, set of values—one for each motivation for each actor). But, first, we note that in most cases costs/benefits are additive, so we don't need to specify the motivation values of performing all combinations of actions A, B, and C. As a simple example, three different actions to purchase items will result in a cost that this the sum of the costs of the things purchased. We specify one cost/benefit per action and sum them based on which actions are taken in a particular outcome. Then, second, we note that in most cases, the ordering of the actions does not affect the costs/benefits. So, instead of having to specify the outcomes of all possible orderings of actions A, B, and C (assuming we are in scenarios where they are all taken), we simply sum the three. So, for instance, the order that you purchase the things you purchase doesn't change the overall cost. The result of these simplifications is that the problem of specifying a scenario is reduced to O(m*n). Then we note that most actions affect a limited number of motivations in most scenarios, so in practice, the specification is closer to O(n) than O(m*n). And as we demonstrated in the previous section, using a graph to represent the action-motivation links that matter is a very efficient way to represent these kinds of sparse matrices.

We do allow for non-additive effects through special cases, though we make the common case easy and rarely find the need to build in exceptions, keeping the process tractable for the scenarios we have analyzed to date. We will discuss these more in the Evaluation section.

Our game solving algorithm is:

1. Generate all possible outcomes (on/off for every action). This step is simply the creation of an array that has 2^n elements where *n* is the number of actions in the scenario. We treat each index into the array as a bit representation of which actions for that entry in the array are on/off. For instance, if there are 3 actions in the scenario, the array will have 8 (2^3) elements that correspond to all possible combinations of actions being of/off. Entry 0 is all off, entry 1 (or 001 in binary) is action 1 on and the others off, entry 2 (or 010 in binary) is action 2 on and the others off, and so forth. Each array entry is an *m*-entry array where *m* is the number of motivations in the scenario and each of the *m* values corresponds to the total cost/benefit value associated with that motivation all of the actions whose bits are set to 1 in the given index value.
2. Eliminate outcomes that are illegal based on constraints. For instance, entry 3 (binary 011) will be marked as not possible if actions 1 and 2 are mutually exclusive.

3. Group the remaining outcomes into sets that are completely controlled by a single actor (that is, where everyone else's actions are fixed) and eliminate dominated outcomes within each set (those the controlling actor will never choose). This is a standard game theory technique for finding pure solutions to games though extended to multi-objective reasoning with ranges of values. In traditional game theory, one outcome is preferred if its utility is higher than other outcomes. In our approach, the outcome is preferred if it is not Pareto-dominated by any other solution. The result is that we find all outcomes that an actor might prefer depending on how they weigh the various costs/benefits. We do not force the user to assume they know how the actors will make those tradeoffs ahead of time, which in the case of cross-cultural analyses is typically appropriate.

Step 1 is simple and fast, but requires an (integer) array that is m * 2ˆn in size, which can potentially be a problem for large scenarios. We have found so far that even complex, real-world scenarios do not cause problems in terms of memory due to the relatively small sizes of m and n in these scenarios. We have also developed techniques that can drastically reduce the memory requirements in many cases where m or n is large, though we do not have the space to go into them in this paper.

This approach can, especially for large scenarios, produce many possible outcomes. We believe this is a benefit of the approach and have developed tools to analyze the outcomes produced. Analyzing the possible tradeoffs each actor faces is a powerful way to deepen understanding of the possible space of real-world outcomes. Any tool that came up with a single optimal solution to scenarios like what might happen in the South China Sea would be difficult to believe and would lack this additional insight.

EVALUATION

To evaluate the flexibility of this approach, we have used our software to create and analyze several very different models, from simple games like Rock-Paper-Scissors and the Prisoners' Dilemma to real-world gray-zone conflicts. The largest model we have built to date is based on the Scarborough Shoal/Spratly Islands conflict between China, the US, and the Philippines in the South China Sea as described in (Corr, 2018). This model includes 3 nation states, 23 possible actions, 18 different motivations, and 10ˆ21 possible solutions if we were to accept all possible action orderings as separate possible solutions. The results were computed in seconds on a laptop and the roughly 50 outcomes that were identified are consistent with the actual and plausible behaviors of the real-world actors.

Next, to ensure that the underlying modeling formalisms were user-friendly enough to be used by our intended end users, we worked with international policy experts to have them develop a series of scenarios with our software. We chose policy experts who are experts in gray zone conflicts but who have limited (if any) computational modeling experience. The goal was to get users of our software who could generate scenarios of interest to the types of analysts we hope will eventually use our software to analyze real-world conflicts. They developed 5 scenarios of their choosing, including: 1938 German Annexation of Czechoslovakia, Defending a Treaty Partner from Gray Zone Aggression, Russia Versus NATO and Lithuania, a 2007 Estonia Cyber Attack, and a dispute between the fictional countries of Atropia and Donovia based on US Army training scenarios. In all cases, the analysts were able to successfully build the models they set out to build. They also provided us feedback on ways to improve the usability of the tool and supporting documentation, many of which we have since added to the prototype and associated training materials.

We also performed a more thorough "backcasting" analysis of the 2007 Estonia cyber attack scenario by looking to see if the results were compatible with the actual outcome of the scenario. In April 2007, the Estonian government decided to move a Soviet World War Two (WWII) memorial statue out of Tallinn's city center. This action sparked protests within Russia. Shortly after, Estonia suffered a series of crippling cyberattacks against bank, government, news, and other websites. Our policy experts developed a simple model of this situation with Russia, Estonia, and NATO able to choose from two to four options and they provided it to us to analyze. Aside from two missing "mutually exclusive" constraints, the model they produced captured the experts' intent. Analysis of that model by our solver suggested one plausible outcome given the situation: Estonia removes the WWII memorial; Russia generally punishes Estonia for the insult, attacks Estonia in a non-traditional way (cyber), and engages proxy actors; and NATO sets up a cyber center in Estonia. This aligns with how things played out in reality. The model was simple and certainly omitted other options for all sides that would likely have resulted in more possible outcomes if built by a more experienced modeler, but the analyst was able to recreate the logic of the situation and our solver was able to find a solution consistent with the motivations of all of the actors.

CONCLUSION

We have developed a novel form of game theory and associated software that simplifies the process of defining a game and analyzing the plausible outcomes in complex, real-world scenarios. The model building tool helps analysts capture the goals and motivations of each actor, the actions available, and how those actions affect goals or other actions. Using these models, the analysis suite calculates the Pareto-optimal choices for each actor in that scenario and helps analysts navigate the plausible outcomes. With these tools, decision makers can assess the value of their strategic options, even in cases where adversaries may choose actions traditional game theory would label incorrect. Based on our preliminary evaluations, we believe that this approach to modeling and analyzing gray zone conflicts is a promising approach with real-world applications and benefits.

ACKNOWLEDGEMENT

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. D17PC00153 and by the

United States Air Force under Contract No. FA8650-19-C-6958. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, its Contracting Agent (the U.S. Department of the Interior, Interior Business Center, Acquisition Services Directorate, Division III), or the United States Air Force. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

REFERENCES

Corr, A. (Ed.). (2018). *Great Powers, Grand Strategies: The New Game in the South China Sea*. Annapolis: Naval Institute Press.

Luce, R. Duncan*;* Raiffa, Howard (1957). *Games and decisions: introduction and critical survey, New York:* Wiley.

Nash, John. (1950). *"Equilibrium points in n-person games", Proceedings of the National Academy of Sciences of the United States of America, 36 (1): 48–49.*

Simon, Herbert A. (1955). "A Behavioral Model of Rational Choice", Quarterly Journal of Economics, 69(1): 99–118.

von Neumann, John*;* Morgenstern, Oskar (1944). "Theory of games and economic behavior", *Nature,* Princeton University Press*, 157 (3981).*