Action-Graph Games

Albert Xin Jiang* Kevin Leyton-Brown† Navin A.R. Bhat‡

Abstract

Representing and reasoning with games becomes difficult once they involve large numbers of actions and players, because utility functions can grow unmanageably. Action-Graph Games (AGGs) are a fully-expressive game representation that can compactly express utility functions with structure such as context-specific (or strict) independence, anonymity, and additivity. We show that AGGs can be used to compactly represent all games that are compact when represented as graphical games, symmetric games, anonymous games, congestion games, and polymatrix games. We further show that AGGs can compactly represent additional, realistic games that require exponential space under all of these existing representations. We give a dynamic programming algorithm for computing a player's expected utility under an arbitrary mixed-strategy profile, which can achieve running times polynomial in the size of an AGG representation. We show how to use this algorithm to achieve exponential speedups of existing methods for computing sample Nash and correlated equilibria. Finally, we present the results of extensive experiments, showing that using AGGs leads to a dramatic increase in the size of games accessible to computational analysis.[1]

Keywords: game representations, graphical models, large games, computational techniques, Nash equilibria.

JEL classification codes: C63—Computational Techniques, C72—Noncooperative Games.

1 Introduction

Simultaneous-action games have received considerable study, which is reasonable as these games are in a sense the most fundamental. Most of the game theory literature presumes that simultaneous-action games will be represented in normal form. This is problematic because in many domains of interest the number of players and/or the number of actions per player is large. In the normal form representation, the game's payoff function is stored as a matrix with one entry for each player's payoff under each combination of all players' actions. As a result, the size of the representation grows exponentially with the number of players.

Fortunately, most large games of practical interest have highly-structured payoff functions, and thus it is possible to represent them compactly. Intuitively, this helps to explain why people are able to reason about these games in the first place: we understand the payoffs in terms of simple relationships rather than in terms of enormous lookup tables. One thread of recent work in the literature has explored game representations that are able to succinctly describe games of interest. In some sense, nearly every game form besides the normal form itself can be seen as

*Department of Computer Science, University of British Columbia.

such a compact representation. For example, the extensive form allows games with temporal structure to be encoded in exponentially less space than the normal form. In what follows, however, we concentrate on game representations that are compact even for simultaneous-move games of perfect information.

Perhaps the most influential class of compact game representations is those that exploit strict independencies between players' utility functions. This class includes graphical games [Kearns *et al.*, 2001; Kearns, 2007], multi-agent influence diagrams [Koller & Milch, 2003], and game nets [LaMura, 2000]; we focus on the first of these. Consider a graph in which nodes correspond to agents and an edge from one node to another represents the proposition that the first agent is able to affect the second agent's payoff. If every node in the graph has a small in-degree—that is, if each agent's payoff depends only on the actions of a small number of others—then the graphical game representation is compact, by which we mean that it is exponentially smaller than its induced normal form. Of course, there are any number of ways of representing games compactly. For example, games of interest could be assigned short ID numbers. What makes graphical games important is the fact that computational questions about these games can be answered by algorithms whose running time depends on the size of the representation rather than the size of the induced normal form. (Note that this property does not hold for the naive ID number scheme.) To state one fundamental property [Daskalakis *et al.*, 2006a], it is possible to compute an agent's expected utility under an arbitrary mixed strategy profile in time polynomial in the size of the graphical game representation. This property implies that a variety of algorithms for computing game-theoretic quantities of interest, such as sample Nash [Govindan & Wilson, 2003; van der Laan *et al.*, 1987] and correlated equilibrium, can be made exponentially faster for graphical games without introducing any change in the algorithms' behavior or output [Blum *et al.*, 2006; Papadimitriou, 2005]. Furthermore, graphical games are also computationally well-behaved in other ways; efficient algorithms exist for computing other quantities of interest for these games such as Nash equilibria on restricted graphs [Kearns *et al.*, 2001; Elkind *et al.*, 2006] or subject to a fairness criterion [Elkind *et al.*, 2007], pure Nash equilibrium [Daskalakis & Papadimitriou, 2006], ϵ-Nash equilibrium [Kearns *et al.*, 2001; Vickrey & Koller, 2002], and evolutionary stable strategies [Kearns & Suri, 2006].

A drawback of the graphical games representation is that it only helps when there exist agents who *never* affect some other agents' utilities. Unfortunately, many games of interest lack any structure of this kind. For example, nontrivial symmetric games are cliques when represented as graphical games. Another useful form of structure not generally captured by graphical games is dubbed *anonymity*; it holds when each agent's utility depends only on the number of agents who took each action, rather than on these agents' identities.[2] Recently, researchers such as Papadimitriou and Roughgarden [2005], Kalai [2005] and Daskalakis and Papadimitriou [2007] have explored the representational, computational and strategic benefits that can be derived from symmetry and anonymity assumptions.

A weaker form of utility independence can usefully be combined with symmetry and anonymity. Specifically, utility functions exhibit *context-specific* independencies when the question of whether two agents are able to affect each other's utilities depends on the actions both agents choose. Congestion games [Rosenthal, 1973] are a prominent game representation that can express context-specific payoff independencies, anonymity, *and* symmetry. The congestion game representation has many advantages. First and most importantly, many realistic interactions—even involving very large numbers of players and actions—have compact representations as congestion games

[2]Note that our definition of anonymity presumes that it makes sense to speak about two different agents having at least some of the same action choices. There are various ways of achieving this formally; for now, one can simply assume that anonymous games are also symmetric.

(see, e.g., [Roughgarden & Tardos, 2002]). Second, congestion games have attractive theoretical properties. Most notably, they always have pure-strategy equilibria, and indeed always admit an exact potential function [Monderer & Shapley, 1996]. As a consequence, simple best-response dynamics are guaranteed to converge to a pure-strategy equilibrium. Finally, congestion games have attractive computational properties. For example, correlated equilibrium can be efficiently computed for congestion games [Papadimitriou, 2005], and pure-strategy Nash equilibrium can be efficiently computed for restricted subclasses of congestion games (see, e.g., [Ieong *et al.*, 2005]).

Unfortunately, congestion games too have a catch. Unlike graphical games, congestion games are not a universal game representation: not every normal-form game can be encoded as a congestion game. Indeed, this problem should be obvious from the fact that congestion games always have pure-strategy equilibria. Congestion games require that agents' utility functions must be expressible as a *sum* of arbitrary functions of the numbers of agents who chose each of a set of resources, where each action is interpreted as the choice of one or more resources. This linearity assumption is restrictive. Thus, while congestion games constitute a useful model for reasoning about certain game-theoretic domains, they cannot serve as the basis for a set of general tools for representing and reasoning about games.

Action-graph games (AGGs) are a general game representation that can be understood as offering the advantages of—and, indeed, unifying—both graphical games and congestion games. Like graphical games, AGGs can represent any game, and game-theoretic computations can be performed efficiently when the AGG representation is compact. Hence, AGGs offer a general representational framework for game-theoretic computation. Like congestion games, AGGs compactly represent context-specific independence, anonymity, and additivity, though unlike congestion games they do not require the latter. Finally, AGGs can also compactly represent many games that are compact neither as graphical games nor as congestion games.

We begin this paper in Section 2 by defining the basic AGG representation, characterizing its representation size, and showing how it can be used to represent normal-form, graphical, and symmetric games. In Section 3 we introduce the idea of *function nodes*, show how this representational device can capture additional structure in several example games, and show how to represent anonymous games as AGGs. Section 4 describes how to represent additive structure in the utility functions of AGGs, and shows how congestion and polymatrix games can be succinctly written as AGGs. Then we turn from representational to computational issues. In Section 5 we present a dynamic programming algorithm for computing an agent's expected utility under an arbitrary mixed-strategy profile, prove its correctness and complexity, and explore several elaborations. In Section 6 we prove that the problem of finding a Nash equilibrium of an AGG is PPAD-complete (a positive result, as AGGs can be exponentially smaller than normal-form games), and show how to use our dynamic programming algorithm to speed up existing methods for computing sample Nash and correlated equilibria. Finally, in Section 7 we present the results of extensive experiments with some of these algorithms, confirming our theoretical predictions and demonstrating that AGGs can feasibly be used to reason about interesting games that were inaccessible to any previous techniques. The largest game that we tackled in our experiments had 20 agents and 13 actions per agent; we found its Nash equilibrium in 14.3 minutes. A normal form representation of this game would involve 9.4×10^{134} numbers, requiring an outrageous 7.5×10^{126} gigabytes even to store.

Finally, let us describe the relationship between this paper and past work, mostly our own, on AGGs. Leyton-Brown and Tennenholtz [2003] introduced local-effect games, which can be understood as symmetric AGGs in which utility functions are required to satisfy a particular linearity property. Bhat and Leyton-Brown [2004] introduced the basic AGG representation and