Supplementary Material for "Bayesian Action-Graph Games": Proofs

Albert Xin Jiang
Department of Computer Science
University of British Columbia
jiang@cs.ubc.ca

Kevin Leyton-Brown
Department of Computer Science
University of British Columbia
kevinlb@cs.ubc.ca

1 Proof of Lemma 4

Proof. Given an arbitrary Bayesian game $(N, \{A_i\}_{i\in N}, \Theta, P, \{u_i\}_{i\in N})$ represented in Bayesian normal form, we construct the BAGG $(N, \Theta, P, \{A'_{i,\theta_i}\}_{i\in N,\theta_i\in\Theta_i}, G, \{u^\alpha\}_{\alpha\in\mathcal{A}})$ as follows. The Bayesian normal form's tabular representation of type profile distribution P can be straightforwardly represented as a BN, e.g. by creating a random variable representing θ as the only parent of the random variables $\theta_1, \ldots, \theta_n$. To represent utility functions, we create an action graph G with $\sum_i |\Theta_i||A_i|$ action nodes; in other words, all type-action sets A'_{i,θ_i} are disjoint. Each action $a_i \in A_i$ of the Bayesian normal form corresponds to $|\Theta_i|$ action nodes in the BAGG, one for each type instantiation θ_i. For each player i and each type $\theta_i \in \Theta_i$, each action node $\alpha \in A'_{i,\theta_i}$ has incoming edges from all action nodes from type-action sets A'_{j,θ_j} for all $j \neq i$, $\theta_j \in \Theta_j$, i.e. all action nodes of the other players. For each action node $\alpha \in A'_{i,\theta_i}$ corresponding to $a_i \in A_i$, the utility function u^α is defined as follows: given configuration $c^{(\alpha)}$ we can infer the action profile $a'_{-i} \in A'_{-i}$ of the BAGG, which then tells us the corresponding a_{-i} and θ_{-i} of the Bayesian normal form, which gives us the utility $u_i(a, \theta)$. The number of utility values stored in this BAGG is the same as the Bayesian normal form. □

2 Proof of Theorem 9

Proof. We reduce the problem of computing expected utility $u_i(\sigma^{\theta_i\to a_i}|\theta_i)$ for BAGGs with independent type distributions to the problem of computing expected utility for AGGs.

Given a BAGG $(N, G, \{u^\alpha\}_{\alpha\in\mathcal{A}})$, we consider the AGG Γ specified by $(N, \{A_i^\cup\}_{i\in N}, G, \{u^\alpha\}_{\alpha\in\mathcal{A}})$, i.e. an AGG with the same set of players, the same action graph and the same utility functions, but with action sets corresponding to total action sets of the BAGG. The representation size of the AGG Γ is proportional to the size of the BAGG. Furthermore, since the BAGG is contribution-independent, all function nodes in the AGG Γ is contribution-independent.

Given i, θ_i and $\sigma^{\theta_i\to a_i}$, for each player $j \neq i$ we can calculate $\Pr(D_j)$ by summing out θ_j: $\Pr(D_j = a_j) = \sum_{\theta_j} \sigma_j(a_j|\theta_j)$. Observe that this distribution of the strategy variable D_j can be interpreted as a (complete-information) mixed strategy σ'_j of the AGG Γ's player j. Similarly for player i, the distribution $\Pr(D_i|\theta_i)$ can be interpreted as a mixed strategy σ'_i of Γ's player i. Furthermore these distributions are independent, so they induce the same distribution over configurations of the BAGG as the distribution over configurations of the AGG Γ induced by the mixed-strategy profile $\sigma' = (\sigma'_1, \ldots, \sigma'_n)$.

Therefore the expected utility $u_i(\sigma^{\theta_i\to a_i}|\theta_i)$ for the BAGG is equal to the expected utility of i in the AGG Γ under the mixed strategy profile σ'. Expected utility for contribution-independent AGGs can be computed in polynomial time by running the algorithm of Jiang and Leyton-Brown [1]. □

An alternative approach for proving Theorem 9 is to work on the TBN of the BAGG, which can be shown to have treewidth as most $|\nu(a_i)|$. Although $|\nu(a_i)|$ is not necessarily a constant so Theorem 8 cannot be directly applied, it can be shown that a variable elimination algorithm needs to store at most $|C^{(a_i)}|$ numbers in each of its tables, which is polynomial in the size of the BAGG. These two proof approaches can be thought of as two interpretations of the same expected utility algorithm.

References

[1] A. X. Jiang and K. Leyton-Brown. A polynomial-time algorithm for Action-Graph Games. In *AAAI*, pages 679–684, 2006.