Algorithm for Evolutionarily Stable Strategies Against Pure Mutations

Sam Ganzfried
Ganzfried Research
sam.ganzfried@gmail.com

Abstract

Evolutionarily stable strategy (ESS) is an important solution concept in game theory which has been applied frequently to biology and even cancer. Finding such a strategy has been shown to be difficult from a theoretical complexity perspective. Informally an ESS is a strategy that if followed by the population cannot be taken over by a mutation strategy. We present an algorithm for the case where mutations are restricted to pure strategies. This is the first positive result for computation of ESS, as all prior results are computational hardness and no prior algorithms have been presented.

1 Introduction

While Nash equilibrium has emerged as the standard solution concept in game theory, it is often criticized as being too weak: often games contain multiple Nash equilibria (sometimes even infinitely many), and we want to select one that satisfies other natural properties. For example, one popular concept that refines Nash equilibrium is evolutionarily stable strategy (ESS). A mixed strategy in a two-player symmetric game is an evolutionarily stable strategy if, informally, it is robust to being overtaken by a mutation strategy. Formally, mixed strategy x^* is an ESS if for every mixed strategy x that differs from x^*, there exists $\epsilon_0 = \epsilon_0(x) > 0$ such that, for all $\epsilon \in (0, \epsilon)$,

$$(1-\epsilon)u_1(x, x^*) + \epsilon u_1(x, x) < (1-\epsilon)u_1(x^*, x^*) + \epsilon u_1(x^*, x). \quad (1)$$

From a biological perspective, we can interpret x^* as a distribution among "normal" individuals within a population, and consider a mutation that makes use of strategy x, assuming that the proportion of the mutation in the population is ϵ. In an ESS, the expected payoff of the mutation is smaller than the expected payoff of a normal individual, and hence the proportion of mutations will decrease and eventually disappear over time, with the composition of the population returning to being mostly x^*. An ESS is therefore a mixed strategy of the column player that is immune to being overtaken by mutations. ESS was initially proposed by mathematical biologists motivated by applications such as population dynamics (e.g., maintaining robustness to mutations within a population of humans or animals) [11, 12]. A common example game is the 2x2 game where strategies correspond to an "aggressive" Hawk or a "peaceful" Dove strategy. A paper has recently proposed a similar game in which an aggressive malignant cell competes with a passive normal cell for biological energy, which has applications to cancer eradication [6].

While Nash equilibrium is defined for general multiplayer games, ESS is defined specifically for two-player symmetric games. ESS is a refinement of Nash equilibrium. In particular, if x^* is an ESS, then (x^*, x^*) (i.e., the strategy profile where both players play x^*) is a (symmetric) Nash equilibrium [10]. Of course the converse is not necessarily true (not every symmetric Nash equilibrium is an ESS), or else ESS would be a trivial refinement. In fact, ESS is not guaranteed to exist in games with more than two pure strategies per player (while Nash equilibrium is guaranteed to exist in all finite games [13]). For example, while rock-paper-scissors has a mixed strategy Nash equilibrium (which puts equal weight on all three actions), it has no ESS [10] (that work considers a version where the payoffs are 1 for a victory, 0 for loss, and $\frac{2}{3}$ for a tie).

There exists a polynomial-time algorithm for computing Nash equilibrium in two-player zero-sum games, while for two-player general-sum and multiplayer games computing a Nash equilibrium is PPAD-complete [4, 3] and it is widely conjectured that no efficient (polynomial-time) algorithm exists. However, several algorithms have been devised that perform well in practice [1, 15, 8, 16, 9]. For ESS, there have been some recent hardness results as well (note that in general computing an ESS is at least as hard as computing a Nash equilibrium since it is a refinement). The problem of computing whether a game has an ESS was shown to be both NP-hard and CO-NP hard and also to be contained in Σ_2^P (the class of decision problems that can be solved in nondeterministic polynomial time when given access to an NP oracle) [7]. Subsequently it was shown that the exact complexity of this problem is that it is Σ_2^P-complete [5]. Note that this result is for the complexity of determining whether an ESS exists in a given game (as discussed above there exist games which have no ESS), and not for the complexity of actually computing an ESS in games for which one exists. Furthermore I am not aware of any algorithms or heuristics for approximating an ESS (while several exist for Nash equilibrium despite computational hardness results). In this work we describe the first positive result for ESS computation. We present an algorithm that computes an ESS that satisfies a less restrictive condition that only pure strategy mutations cannot successfully invade the population (while standard ESS rules out all mixed strategy mutations). The algorithm is based on modeling the problem as a mixed-integer non-convex quadratically-constrained feasibility program.

2 Evolutionarily stable strategies against pure mutations

We define an *Evolutionarily Stable Strategy against Pure Mutations* (ESSPM) as in Equation 1 except that we only require that the inequality holds for all pure strategies x that differ from x^*. In order to model the problem of computing an ESSPM as an optimization problem, we recall a widely-known alternative definition of ESS that has been proven to be equivalent to the initial one [10].

Definition 1. *A mixed strategy x^* in a two-player symmetric game is an evolutionarily stable strategy (ESS) if for every mixed strategy x that differs from x^* there exists $\epsilon_0 = \epsilon_0(x)$ such that, for all $\epsilon \in (0, \epsilon_0)$,*

$$(1-\epsilon)u_1(x, x^*) + \epsilon u_1(x, x) < (1-\epsilon)u_1(x^*, x^*) + \epsilon u_1(x^*, x).$$

Theorem 1. *A strategy x^* is evolutionarily stable if and only if for each $x \neq x^*$ exactly one of the following conditions holds:*

- $u_1(x, x^*) < u_1(x^*, x^*)$
- $u_1(x, x^*) = u_1(x^*, x^*)$ *and* $u_1(x, x) < u_1(x^*, x)$

Definition 2. *A mixed strategy x^* in a two-player symmetric game is an evolutionarily stable strategy against pure mutations (ESSPM) if for every pure strategy x that differs from x^* there exists $\epsilon_0 = \epsilon_0(x)$ such that, for all $\epsilon \in (0, \epsilon_0)$,*

$$(1-\epsilon)u_1(x, x^*) + \epsilon u_1(x, x) < (1-\epsilon)u_1(x^*, x^*) + \epsilon u_1(x^*, x). \quad (2)$$

Theorem 2. *A strategy x^* is evolutionarily stable against pure mutations if and only if for each pure strategy x exactly one of the following conditions holds:*

- $u_1(x, x^*) < u_1(x^*, x^*)$

- $u_1(x, x^*) = u_1(x^*, x^*)$ *and* $u_1(x, x) < u_1(x^*, x)$

The proof of Theorem 2 follows from similar reasoning as the proof of Theorem 1 [10].

There are also known results that allow us to categorize ESS with respect to the common solution concept of Nash equilibrium [10]:

Theorem 3. *If x^* is a an evolutionarily stable strategy in a two-player symmetric game, then (x^*, x^*) is a symmetric Nash equilibrium of the game.*

Theorem 4. *In a symmetric game, if (x^*, x^*) is a strict symmetric Nash equilibrium then x^* is an evolutionarily stable strategy.*

Recall that in a Nash equilibrium x^*, we have that $u_1(x, x^*) \leq u_1(x^*, x^*)$ for all mixed strategies x (and similarly for the other players). In a strict equilibrium this requirement changes to $u_1(x, x^*) < u_1(x^*, x^*)$.

Clearly every ESS is also an ESSPM, and so therefore ESS is a refinement of ESSPM. We can straightforwardly show a similar result to Theorem 3 for ESSPM. This shows that ESSPM is a refinement of Nash equilibrium.

Theorem 5. *If x^* is a an ESSPM in a two-player symmetric game, then (x^*, x^*) is a symmetric Nash equilibrium of the game.*

Proof. Consider taking the limit as $\epsilon \to 0$ in Equation 2. From the continuity of the utility function (which is a standard assumption in game theory), this gives $u_1(x, x^*) \leq u_1(x^*, x^*)$ for all pure strategies x. This condition is sufficient to show that x^* is a Nash equilibrium; if a player could profitably deviate from x^* to a mixed strategy x', then for all the pure strategies in the support of x' he would obtain higher payoff against x^* than by following x^*, which contradicts the fact that $u_1(x, x^*) \leq u_1(x^*, x^*)$ for all pure strategies. So (x^*, x^*) is a Nash equilibrium, and is symmetric because the strategies for the players are the same. □

We can also show a result similar to Theorem 4 for ESSPM.

Theorem 6. *In a symmetric game, if (x^*, x^*) is a strict symmetric Nash equilibrium then x^* is an ESSPM.*

Proof. If (x^*, x^*) is a strict symmetric Nash equilibrium, then x^* satisfies the first condition from Theorem 2 for all mixed strategies, and therefore also satisfies the condition for all pure strategies x. Therefore, x^* is an ESSPM. □

3 Algorithm

Given the formulation from Theorem 2, we can cast the problem of computing an ESSPM as the following feasibility program:

$$u_1(x, x^*) \leq u_1(x^*, x^*) - \epsilon_{x1} + M_{x1} y_x \text{ for all } x \in [1, \ldots, m] \quad (3)$$

$$u_1(x, x^*) \leq u_1(x^*, x^*) + M_{x2}(1 - y_x) \text{ for all } x \in [1, \ldots, m] \quad (4)$$

$$u_1(x^*, x^*) \leq u_1(x, x^*) + M_{x3}(1 - y_x) \text{ for all } x \in [1, \ldots, m] \quad (5)$$

$$u_1(x, x) \leq u_1(x^*, x) - \epsilon_{x2} + M_{x4}(1 - y_x) \text{ for all } x \in [1, \ldots, m] \quad (6)$$

$$x^*_i \geq 0 \text{ for all } i \in [1, \ldots, m] \quad (7)$$

$$\sum_i x^*_i = 1 \quad (8)$$

$$y_i \text{ binary for all } i \in [1, \ldots, m] \quad (9)$$

We assume that we are initially given an $m \times m$ matrix A of utilities to player 1 (we assume that the game is symmetric since ESS is defined only for symmetric games, so payoffs to player 2 are given by matrix $B = A^T$). If u and v are pure strategies, the expression $u_1(u, v)$ corresponds to the u'th row and v'th column of matrix A. In general if u and v are vectors of mixed strategies (i.e., probability distributions over pure strategies), then the utility for player 1 is given by $u_1(v, w) = v^T Aw$. Without loss of generality we can assume that all values in A are nonnegative and between 0 and 1 (note that any affine transformation does not affect strategic aspects of the game, so we can add a sufficiently large constant to all entries and normalize by dividing by the largest entry to achieve this condition).

The variables in the formulation are x_i^* and y_i for $1 \leq i \leq m$. The x_i^* correspond to the ESSPM for player 1 (and equivalently for player 2) that we are seeking to compute, and y_i correspond to indicator variables denoting which of the conditions holds from Theorem 2 for the given component.

We set ϵ to be a very small floating point number slightly larger than 0, such as $\epsilon = 0.00001$. M denotes a constant that exceeds the maximum difference in absolute value of utility between two strategy profiles. We would prefer to have M be as small as possible to make the inequality tighter, so we will set $M = 1 + \epsilon$, given our assumption that all payoffs are between 0 and 1. Note that our framework is flexible enough to allow for different selections for the different ϵ_{xi} and M_{xi} parameters corresponding to the different constraints if desired, though for simplicity we set them all to be the same ϵ and M as described. The equivalence of the feasibility program to the conditions of Theorem 2 follows from a rule for representing either-or constraints in a mixed-integer program by adding in auxiliary binary indicator variables [2]. If $y_x = 0$, then the first constraint ensures that $u_1(x, x^*) \leq u_1(x^*, x^*) - \epsilon_{x1}$, which is equivalent to $u_1(x, x^*) < u_1(x^*, x^*)$ since ϵ_{x1} is negligible (note that strict inequalities must be converted to weak inequalities to be solved by most standard optimization algorithms). And if $y_x = 1$ then the constraint states $u_1(x, x^*) \leq u_1(x^*, x^*) - \epsilon_{x1} + M_{x1}$, which essentially makes the constraint inactive since it will be true for all strategies x^* since we have chosen M_{x1} to be sufficiently large. The translation between the other constraints and the theorem can be obtained similarly. Note that this formulation is a quadratically-constrained mixed-integer feasibility program (QCMIP). It is quadratically constrained because $u_1(x^*, x^*)$ equals $(x^{*T} Ax^*)$, which is a quadratic form involving a product of the variables x^*. It is mixed integral because there are both continuous and binary variables. And it is a feasibility program because there is no objective function, though there are several natural objective functions that could potentially be used that may help aid performance in practice.

We must apply a few additional steps to make this formulation practical. Assume that we have a QCMIP solving algorithm available to apply. For our implementation we use algorithms from the QVXPY software package [14]. These algorithms are heuristic and have no performance guarantee, though have been demonstrated to perform well empirically on certain benchmark problems. As stated earlier a game may have no ESS, so there is no guarantee that a solution to the ESSPM formulation will exist. While the QVXPY algorithms have strong performance in general, there is no guarantee that solution quality will improve monotonically or that it converge (note that we could measure solution quality for our problem in terms of the degree to which the constraints are violated since there is no objective function). Thus the solution generated at each algorithm iteration may only measure a subset of the constraints. Our overall meta-algorithm is given by Algorithm 1. It assumes a QCMIP algorithm A as input, as well as values for the algorithm parameters M_{ij}, ϵ_{ij} and a desired number of iterations T. At each iteration it solves the ESSPM with the given parameters. Since it is possible that the solution produced by the algorithm violates the constraints that $0 \leq x_i^* \leq 1$ and $\sum_i x_i^* = 1$, we will first normalize the entries to ensure feasibility. We will next compute $\delta_t = \max_x u_1(x, x^*) - u_1(x^*, x^*)$ to see the maximum improvement in utility player 1 could obtain by playing pure strategy x as opposed to x^*. Then after T iterations, we will return the strategy that produced the lowest value of δ_t over all the iterations (the lowest value may have been obtained at an earlier iteration since the algorithms are not guaranteed to improve monotonically).

We implemented the algorithm using the CVXPY non-convex quadratically-constrained quadratic pro-

Algorithm 1 ESS computation algorithm

Inputs: QCMIP algorithm A, constant parameters M_{ij}, ϵ_{ij}, number of iterations T

for $t = 1$ to T **do**

Solve ESSPM formulation with parameters M_{ij}, ϵ_{ij} using algorithm A to obtain solution x_t^*.

Normalize x_t^* so that all entries are nonnegative and sum to 1.

$\delta_t \leftarrow$ max improvement in utility by a pure strategy against the normalized x_t^* strategy.

$t^* \leftarrow$ iteration that produces lowest value of δ_t.

Return strategies from iteration t^* with corresponding degree of approximation error of δ_{t^*}.

gram (QCQP) Suggest-and-Improve framework [14]. We chose the coordinate-descent algorithm, because it focuses more on trying to find and maintain feasibility as opposed to the other algorithms which are more focused on optimality. This is appropriate for our problem since it is a feasibility program with no objective function. We create an initial solution for the algorithm by solving the semidefinite program (SDP) relaxation, which we then attempt to improve iteratively using coordinate descent. For the parameters we set $\epsilon_{xi} = 0.00001$ and $M_{xi} = 1 + \epsilon_{xi}$.

We tested our approach on the well-studied Mutation-Population game (Figure 1) [10]. In this game there are two types of animals: hawks (who are aggressive), and doves (who are peaceful). When an animal invades the territory of another animal of the same species, a hawk will aggressively repel the invader, while a dove will yield and be driven out of the territory. The game has one symmetric Nash equilibrium in which Dove is selected with probability $\frac{1}{5}$ and Hawk $\frac{4}{5}$. This strategy profile also constitutes an ESS and an ESSPM. (Note that the game also has two asymmetric Nash equilibria (Dove, Hawk) and (Hawk, Dove), both of which are neither ESS nor ESSPM.)

	Dove	Hawk
Dove	(4,4)	(2,8)
Hawk	(8,2)	(1,1)

Figure 1: The Mutation-Population game

When we ran our algorithm on this game, the initial solution obtained from the SDP relaxation was (0.637, 0.363). and the after the first iteration of the algorithm was (0.245, 0.755), which is quite close to the ESSPM. However, the solution varied drastically between iterations as the algorithm progressed and did not appear to converge over 1000 iterations, though the degree of violation of the program constraints seems to quite small and generally decreasing over the algorithm iterations. We plan to run extensive experiments in the future to ascertain the performance of our algorithm and consider improvements to the implementation than can lead to strong performance (we note that convergence is not guaranteed for algorithms for this problem, though good performance has been observed for previous benchmark domains [14]).

4 Conclusion

In this brief article we have presented an optimization formulation and algorithm that constitutes the first positive result for computation of Evolutionarily Stable Strategies (ESS) (previously the known computational results were all negative hardness results). ESS is a well-studied refinement of Nash equilibrium that is biologically-motivated and has been applied to cancer. We consider a restriction where robustness is guaranteed only against pure-strategy mutations and not necessarily against all mixed-strategy mutations, which we call Evolutionarily Stable Strategies against Pure Mutations (ESSPM). The algorithm is based on a new quadratically-constrained mixed integer feasibility program formulation.

We would like to run comprehensive experiments to determine the scalability and the performance of our algorithm. We would like to start by solving randomly generated $m \times m$ games, and create plots of running time and solution quality as a function of m. We would also like to explore some of the other QCQP-solving algorithms and initialization procedures from CVXPY in addition to coordinate descent, as well as experiment with objective functions that can help guide the optimization search even though they are not required in our formulation. We would also like to consider the general problem of computing an ESS that is robust to all mutation strategies, not just to pure strategy mutations as we have done.

References

[1] Kimmo Berg and Tuomas Sandholm. Exclusion method for finding nash equilibrium in multiplayer games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[2] Johannes Bisschop. *AIMMS—Optimization Modeling*. Paragon Decision Technology, 2006.

[3] Xi Chen and Xiaotie Deng. 3-Nash is PPAD-complete. *Electronic Colloquium on Computational Complexity*, Report No. 134, 2005.

[4] Xi Chen and Xiaotie Deng. Settling the complexity of 2-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, 2006.

[5] Vincent Conitzer. The exact computational complexity of evolutionarily stable strategies. In *Conference on Web and Internet Economics (WINE-13)*, pages 96–108, 2013.

[6] F A C C D Dingli, F C Santos Chalub, S Van Segbroek, and J M Pacheco. Cancer phenotype as the outcome of an evolutionary game between normal and malignant cells. *British Journal of Cancer*, 101(7):1130–1136, 2009.

[7] Kousha Etessami and Andreas Lochbihler. The computational complexity of evolutionarily stable strategies. *International Journal of Game Theory*, 37(1):93–113, 2008.

[8] Srihari Govindan and Robert Wilson. A global Newton method to compute Nash equilibria. *Journal of Economic Theory*, 110:65–86, 2003.

[9] Carlton Lemke and J. Howson. Equilibrium points of bimatrix games. *Journal of the Society of Industrial and Applied Mathematics*, 12:413–423, 1964.

[10] Michael Maschler, Eilon Solan, and Shmuel Zamir. *Game Theory*. Cambridge University Press, 2013.

[11] J Maynard Smith and GR Price. The logic of animal conflict. *Nature*, 2461:13–18, 1973.

[12] John Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[13] John Nash. *Non-cooperative games*. PhD thesis, Priceton University, 1950.

[14] Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870 [math.OC]*, 2017.

[15] Ryan Porter, Eugene Nudelman, and Yoav Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008.

[16] Tuomas Sandholm, Andrew Gilpin, and Vincent Conitzer. Mixed-integer programming methods for finding Nash equilibria. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 495–501, 2005.