Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Algorithms for Strong Nash Equilibrium with More than Two Agents

Nicola Gatti and **Marco Rocco**
Politecnico di Milano
Piazza Leonardo da Vinci 32
Milano, Italy
{ngatti, mrocco}@elet.polimi.it

Tuomas Sandholm
Carnegie Mellon University
Computer Science Department
5000 Forbes Avenue
Pittsburgh, PA 15213, USA
sandholm@cs.cmu.edu

Abstract

Strong Nash equilibrium (SNE) is an appealing solution concept when rational agents can form coalitions. A strategy profile is an SNE if no coalition of agents can benefit by deviating. We present the first general–purpose algorithms for SNE finding in games with more than two agents. An SNE must simultaneously be a Nash equilibrium (NE) and the optimal solution of multiple non–convex optimization problems. This makes even the derivation of necessary and sufficient mathematical equilibrium constraints difficult. We show that forcing an SNE to be resilient only to pure–strategy deviations by coalitions, unlike for NEs, is only a necessary condition here. Second, we show that the application of Karush–Kuhn–Tucker conditions leads to another set of necessary conditions that are not sufficient. Third, we show that forcing the Pareto efficiency of an SNE for each coalition with respect to coalition correlated strategies is sufficient but not necessary. We then develop a tree search algorithm for SNE finding. At each node, it calls an oracle to suggest a candidate SNE and then verifies the candidate. We show that our new necessary conditions can be leveraged to make the oracle more powerful. Experiments validate the overall approach and show that the new conditions significantly reduce search tree size compared to using NE conditions alone.

Introduction

Equilibrium computation in non–cooperative games has recently received significant attention in artificial intelligence and computer science at large. Many papers have focused on the computational study of *Nash equilibrium* (NE) (Shoham and Leyton-Brown 2008), showing that searching for it is $\mathcal{PPAD}$–complete (Daskalakis, Goldberg, and Papadimitriou 2006) even with two agents (Chen, Deng, and Teng 2009) and designing various algorithms. Two–agent games can be solved by linear complementarity programming (Lemke and Howson 1964), support enumeration (Porter, Nudelman, and Shoham 2009), mixed–integer linear programming (Sandholm, Gilpin, and Conitzer 2005), or local search (Gatti et al. 2012). With more agents, common methods are nonlinear complementarity programming, simplicial subdivision, homotopy (Shoham and Leyton-Brown 2008), and support enumeration (Thompson, Leung, and Leyton-Brown 2011).

The *strong Nash equilibrium (SNE)* concept strengthens NE by requiring the strategy profile to be resilient also to multilateral deviations, including deviations by the grand coalition that contains all the agents (Aumann 1960). It captures the situation in which agents can form coalitions and change their strategies multilaterally in a coordinated way. For a given game, an SNE may or may not exist. Searching for it is $\mathcal{NP}$–complete when the number of agents is a constant (Conitzer and Sandholm 2008; Gatti, Rocco, and Sandholm 2013). An SNE must be simultaneously an NE and the optimal solution of multiple non–convex optimization problems (Hoefer and Skopalik 2010). This makes even the derivation of necessary and sufficient mathematical equilibrium constraints a difficult (and currently open) task. Some results have been proven about the computation of pure–strategy SNEs in specific classes of games, e.g., congestion games (Holzman and Law-Yone 1997; Hayrapetyan, Tardos, and Wexler 2006; Rozenfeld and Tennenholtz 2006; Hoefer and Skopalik 2010), connection games (Epstein, Feldman, and Mansour 2007), maxcut games (Gourvès and Monnot 2009), and continuous games (Nessah and Tian 2012). The only prior algorithm that works also for mixed strategies is very recent (Gatti, Rocco, and Sandholm 2013). It is only for 2–agent games. It is a special kind of tree search algorithm, and at each node it calls an $\mathcal{NP}$–complete oracle—a variation of *MIP Nash* (Sandholm, Gilpin, and Conitzer 2005)—that returns an NE (if one exists) in a given subspace of the agents' utilities and then verifies whether the returned NE is an SNE. With more than two agents, MIP Nash cannot be used because the problem of finding an NE is itself already nonlinear.

In this paper, we provide two necessary, but non–sufficient, conditions for the existence of an SNE, conditions that can be used to test whether a game admits an SNE.

- We provide a mixed–integer nonlinear program to find NEs that are resilient to pure–strategy multilateral deviations.
- We provide a nonlinear program to find an NE that satisfies Karush–Kuhn–Tucker conditions (Miettinen 1999).

We also provide a sufficient, but non–necessary, condition

---

for the existence of an SNE that can be used to search for an SNE.

- We provide a nonlinear program in which a strategy profile is forced to be Pareto efficient with respect to coalition correlated strategies.

Then we characterize the relationships between the solutions of these three formulations, NEs, and SNEs.

Finally, we exploit these results to extend the prior SNE–finding tree search algorithm (Gatti, Rocco, and Sandholm 2013) to multiple agents. We do this by introducing a generalization to the tree search framework and by leveraging our necessary conditions in the oracle that is used at each search node. We conduct experiments that validate the overall tree search approach and show the benefits of our new nonlinear necessary conditions in the oracle.

Game–theoretic preliminaries

A strategic–form game (Shoham and Leyton-Brown 2008) is a tuple (N, A, U) where:

- $N = \{1, \ldots, n\}$ is the set of agents (we denote by i a generic agent),
- $A = A_1 \times \ldots \times A_n$ is the set of agents' joint actions, and A_i is the set of agent i's actions (we denote a generic action by a, and by m_i the number of actions in A_i),
- $U = \{U_1 \ldots, U_n\}$ is the set of agents' utility arrays where $U_i(a_1, \ldots, a_n)$ is the utility of agent i when the agents play actions $a_1, \ldots, a_n$.

We denote by $x_i(a_i)$ the probability with which agent i plays action $a_i \in A_i$ and by $\mathbf{x}_i$ the vector of probabilities $x_i(a_i)$ of agent i. We denote by Δ_i the space of well–defined probability vectors over A_i.

An NE is defined as a strategy profile $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ such that, for each $i \in N$, $\mathbf{x}_i^T U_i \prod_{j \neq i} \mathbf{x}_{-j} \geq \mathbf{x}_i'^T U_i \prod_{j \neq i} \mathbf{x}_{-j}$ for every $\mathbf{x}_i' \in \Delta_i$. Every finite game admits at least one NE in mixed strategies, but an NE in pure strategies may or may not exist. The problem of finding an NE can be expressed as:

$$v_i - \sum_{a_{-i} \in A_{-i}} U_i(a_i, a_{-i}) \cdot \prod_{\substack{j \in N: \\ j \neq i}} x_j(a_j) \geq 0 \qquad \begin{array}{l} \forall i \in N, \\ a_i \in A_i \end{array} \tag{1}$$

$$x_i(a_i) \cdot \left(v_i - \sum_{a_{-i} \in A_{-i}} U_i(a_i, a_{-i}) \cdot \prod_{\substack{j \in N: \\ j \neq i}} x_j(a_j) \right) = 0 \qquad \begin{array}{l} \forall i \in N, \\ a_i \in A_i \end{array} \tag{2}$$

$$x_i(a_i) \geq 0 \qquad \begin{array}{l} \forall i \in N, \\ a_i \in A_i \end{array} \tag{3}$$

$$\sum_{a_i \in A_i} x_i(a_i) = 1 \qquad \forall i \in N \tag{4}$$

Here v_i is the expected utility of agent i. Constraints (1) force the expected utility v_i of agent i to be no smaller than the expected utility given by every action a_i available to agent i. Constraints (2) force v_i to be equal to the expected utility given by every action a_i that is played with positive probability by agent i. Constraints (3) force each probability $x_i(a_i)$ to be nonnegative. Constraints (4) force each agent's probabilities to sum to one.

An SNE (Aumann 1960) strengthens the NE concept requiring the strategy profile to be resilient also to multilateral deviations by any coalition of agents. That is, in an SNE no coalition of agents can deviate in a way that strictly increases the expected utility of each member of the coalition, again keeping the strategies of the agents outside the coalition fixed. An SNE combines two concepts: an SNE is an NE and it is weakly Pareto efficient over the space of all the strategy profiles (including the mixed ones) for each possible coalition (fixing the strategies of the agents outside the coalition). Unlike an NE, an SNE may not exist.

Strong Nash equilibrium conditions

We focus on the problem of deriving equilibrium constraints in a mathematical programming fashion for SNE. Denote by $\mathcal{C} = \{N' : N' \subseteq N, |N'| \geq 2\}$ the set of non–singleton coalitions and by $C \in \mathcal{C}$ a generic coalition. Denote by a_C a profile of actions, one for each agent i that is a member of C, where $a_C(i)$ is the action of agent i in a_C, while a_{-C} is defined similarly for the agents that are not members of C.

A strategy profile $\mathbf{x}$ is an SNE if and only if:

- it is an NE, that is, it satisfies constraints (1)–(4), and
- it is Pareto efficient for each $C \in \mathcal{C}$. That is, for each C, it is an optimal solution to the multi–objective mathematical program in which the objectives are the expected utilities of the agents that are members of C:

$$\max_{\mathbf{x}_C} \left[\sum_{(a_C, a_{-C}) \in A} U_i(a_C, a_{-C}) \cdot \cdot x_C(a_C) \cdot \prod_{j \notin C} x_j(a_j) : i \in C \right] \tag{5}$$

$$x_C(a_C) = \prod_{i \in C} x_i(a_C(i)) \qquad \forall a_C \in A_C \tag{6}$$

$$x_i(a_i) \geq 0 \qquad \forall i \in C, a_i \in A_i \tag{7}$$

$$\sum_{a_i \in A_i} x_i(a_i) = 1 \qquad \forall i \in C \tag{8}$$

Here the objective function (5) is the vector of the expected utility functions of the agents i of the coalition C. Constraints (6) force the strategy $\mathbf{x}_C$ of the coalition over the profile a_C to be equal to the product of the strategy $x_i(a_C(i))$ of each single agent i of C. Constraints (7) and (8) force $\mathbf{x}_i$ to be a well–defined strategy.

The above formulations (i.e., NE constraints (1)–(4) and multi–objective programs (5)–(8), one for each C) constitute separate programs that must be satisfied together. Their integration is not straightforward. Consider, e.g., 2–agent games. Only one coalition is possible, i.e., $C = \{1, 2\}$. If we solve program (5)–(8) for $C = \{1, 2\}$ with, as additional constraints, NE constraints (1)–(4), we are searching for NE that is Pareto efficient among all the NEs. Instead, an SNE is an NE that is Pareto efficient among all the strategy profiles. Thus, in order to find equilibrium constraints for SNE, we need to translate, for each C, program (5)–(8) into a feasibility problem that is satisfied by, and only by, all the optimal solutions of (5)–(8). The derivation of such necessary and sufficient equilibrium constraints is an open problem. It is elusive because program (5)–(8) is non–convex and therefore strong duality and complementary slackness conditions do not hold—unlike in the case of NE. In the next three subsections we derive necessary *or* sufficient conditions for SNE using three different ideas, respectively.

NEs resilient to pure multilateral deviations

Our approach in this subsection is to require Pareto optimality of a solution w.r.t. only pure joint strategies (i.e., no coalition can deviate from a strategy profile to a pure–strategy profile that is strictly Pareto dominant for that coalition).

A motivation for this approach comes from what is known for NE. Although the NE concept requires a strategy to be the best w.r.t all mixed strategies, it is sufficient to require that a strategy is best w.r.t. all pure–strategy deviations.

Obviously, resilience to pure multilateral deviations is a necessary condition for an SNE. However, constraining a strategy to be Pareto optimal w.r.t. pure coalition strategies is not straightforward. A unilateral deviation is always possible, i.e., for each strategy, if there is an action that is better than the strategy, the agent will deviate. Instead, a multilateral deviation is possible if and only if all the members of the coalition can gain more by deviating. The need for activating and deactivating constraints related to a multilateral deviation pushes us to resort to mixed integer programming. Given a coalition C, we can formulate the constraints to force an NE to be resilient to pure multilateral deviations of C as a mixed integer nonlinear program:

$$r_{i,C}(a_C) \in \{0,1\} \qquad \begin{matrix}\forall i \in C,\\ a_C \in A_C\end{matrix} \tag{9}$$

$$v_i - \sum_{a_{-C} \in A_{-C}} U_i(a_C, a_{-C}) \cdot \prod_{j \in -C} x_j(a_j) \geq -M \cdot (|C| - \sum_{j \in C} r_{j,C}(a_C)) \qquad \begin{matrix}\forall i \in C,\\ a_C \in A_C\end{matrix} \tag{10}$$

$$v_i - \sum_{a_{-C} \in A_{-C}} U_i(a_C, a_{-C}) \cdot \cdot \prod_{j \in -C} x_j(a_j) \geq -M \cdot r_{i,C}(a_C) \qquad \begin{matrix}\forall i \in C,\\ a_C \in A_C\end{matrix} \tag{11}$$

where M is the largest payoff of all the agents. Constraints (9) force $r_{i,C}(a_C)$ to take on binary values. If the left hand side of (11) is negative, then $r_{i,C}(a_C) = 1$. Constraints (10) force the left hand side to be positive if the variables $r_{i,C}(a_C)$ of all the members of the coalition are 1. That is, if playing a_C is the best for all the members of the coalition, then this multilateral deviation is active and the utility of each member of the coalition must be at least the utility given by playing a_C. Finding an NE that is resilient to pure multilateral deviations can then be formulated as:

constraints (1), (2), (3), (4)
constraints (9), (10), (11) for all $C \in \mathcal{C}$

If there are only two agents, this set of constraints constitutes a mixed integer *linear* program, given that the NE constraints can be expressed in mixed integer linear fashion (Sandholm, Gilpin, and Conitzer 2005).

We can state the following theorem, whose proof easily follows from the derivation above.

Theorem 1 *The above constraints are necessary conditions for a strategy profile to be an SNE.*

However, unlike in the case of NE, they are not sufficient:

Theorem 2 *Resilience to pure multilateral deviations is not sufficient for SNE.*

Proof. Consider the game in Fig. 1. There are three NEs: one pure, (a_3, a_6), and two mixed, $(\frac{1}{2}\mathsf{a}_1 + \frac{1}{2}\mathsf{a}_2, \frac{1}{2}\mathsf{a}_4 + \frac{1}{2}\mathsf{a}_5)$ and $(\frac{1}{7}\mathsf{a}_1 + \frac{1}{7}\mathsf{a}_2 + \frac{5}{7}\mathsf{a}_3, \frac{1}{7}\mathsf{a}_4 + \frac{1}{7}\mathsf{a}_5 + \frac{5}{7}\mathsf{a}_6)$. Focus on $(\mathsf{a}_3, \mathsf{a}_6)$: there is no outcome achievable by pure–strategy multilateral deviations that provides both agents a utility strictly greater than 1. For instance, $(\mathsf{a}_1, \mathsf{a}_4)$ is better for agent 1 than $(\mathsf{a}_3, \mathsf{a}_6)$, but it is not for agent 2. With $(\mathsf{a}_2, \mathsf{a}_4)$ we have the reverse. However, $(\mathsf{a}_3, \mathsf{a}_6)$ is not weakly Pareto efficient, as shown by the Pareto frontier in the figure. Indeed, $(\frac{1}{2}\mathsf{a}_1 + \frac{1}{2}\mathsf{a}_2, \frac{1}{2}\mathsf{a}_4 + \frac{1}{2}\mathsf{a}_5)$ strictly Pareto dominates $(\mathsf{a}_3, \mathsf{a}_6)$. The former, being on the Pareto frontier, is an SNE. □

		agent 2		
		a_4	a_5	a_6
agent 1	a_1	$5, 0$	$0, 5$	$0, 0$
	a_2	$0, 5$	$5, 0$	$0, 0$
	a_3	$0, 0$	$0, 0$	$1, 1$

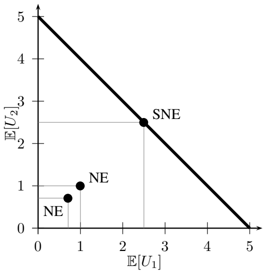

Figure 1: 2–agent game (left) and its Pareto frontier (right).

The above conditions are stronger than the NE conditions (the proof is omitted because it is trivial):

Proposition 3 *There are NEs that do not satisfy constraints (9)–(11).*

NEs satisfying KKT conditions

In this subsection, we present our second formulation approach, which is based on Karush–Kuhn–Tucker (KKT) conditions. Multi–objective programming provides KKT techniques to derive optimality (i.e., Pareto efficiency) conditions. These conditions are:

$$\sum_i \lambda_i \cdot \nabla f_i(\mathbf{z}) + \sum_j \mu_j \cdot \nabla g_j(\mathbf{z}) +$$

$$+ \sum_k \nu_k \cdot \nabla h_k(\mathbf{z}) = \mathbf{0} \qquad (12)$$

$$\mu_j \cdot g_j(\mathbf{z}) = 0 \qquad \forall j \qquad (13)$$

$$\lambda_i, \mu_j \geq 0 \qquad \forall i, j \qquad (14)$$

$$\sum_i \lambda_i = 1 \qquad (15)$$

where $f_i(\mathbf{z})$ are the objective functions to minimize, $g_j(\mathbf{z})$ are inequality constraints of the form $g_j(\mathbf{z}) \leq 0$, h_k are equality constraints of the form $h_k(\mathbf{z}) = 0$. The λ_i, μ_j, ν_k are called KKT multipliers: λ_i is the weight of objective function f_i, μ_j is the weight of constraint g_j, and ν_k is the weight of constraint h_k.

KKT conditions (12)–(15) are necessary conditions for local Pareto efficiency (Miettinen 1999). We can map these conditions to the case of Pareto efficiency for a single coalition C as follows:

- f_i: is agent i's expected utility multiplied by '-1' (given that in KKT f_i is to minimize);
- g_j: is a constraint of the form $-x_w(a_w) \leq 0$;
- h_k: is a constraint of the form $\sum_{a_i \in A_i} x_i(a_i) - 1 = 0$.

Given a coalition C, we obtain the following conditions:

$$-\sum_{i\in C} \lambda_{i,C} \cdot \sum_{\substack{a\in A:\\ a_w=\overline{a}_w}} \prod_{\substack{j\in C:\\ j\neq w}} U_i(a) \cdot x_j(a_j) \cdot$$

$$\cdot \prod_{j \notin C} x_j(a_j) - \mu_C(\overline{a}_w) + \nu_{w,C} = 0 \qquad \begin{array}{l}\forall w \in C,\\ \overline{a}_w \in A_w\end{array} \qquad (16)$$

$$-\mu_C(a_i) \cdot x_i(a_i) = 0 \qquad \begin{array}{l}\forall i \in C,\\ a_i \in A_i\end{array} \qquad (17)$$

$$\lambda_{i,C} \geq 0 \qquad \forall i \in C \qquad (18)$$

$$\mu_C(a_i) \geq 0 \qquad \begin{array}{l}\forall i \in C,\\ a_i \in A_i\end{array} \qquad (19)$$

$$\sum_{i \in C} \lambda_{i,C} = 1 \qquad (20)$$

Now, we can leverage the above results, producing a non-linear mathematical program for SNE as follows:

constraints (1), (2), (3), (4)
constraints (16)–(20) $\forall C \in \mathcal{C}$

We can state the following theorem, whose proof easily follows from the derivation above.

Theorem 4 *The above constraints are necessary conditions for a strategy profile to be an SNE.*

However, such conditions are not sufficient.

Theorem 5 *The above constraints are not sufficient for a strategy profile to be an SNE, nor for NEs to be locally Pareto efficient.*

Proof. Consider the game in Fig. 2. The game admits a mixed strategy NE $(\frac{1}{2}a_1 + \frac{1}{2}a_2, \frac{1}{2}a_5 + \frac{1}{2}a_6)$ that gives each of the two agents utility 1. This NE is Pareto dominated by, e.g., $(\frac{1}{2}a_3 + \frac{1}{2}a_4, \frac{1}{2}a_7 + \frac{1}{2}a_8)$ that gives each agent utility $\frac{9}{4}$. The previous NE satisfies KKT conditions for all feasible λ_1, λ_2 with $\mu_1(a_3) = \mu_1(a_4) = \mu_2(a_7) = \mu_2(a_8) = 6$ and $\nu_1 = \nu_2 = 1$. ☐

		agent 2			
		a_5	a_6	a_7	a_8
agent 1	a_1	$2, 0$	$0, 2$	$-5, -5$	$-5, -5$
	a_2	$0, 2$	$2, 0$	$-5, -5$	$-5, -5$
	a_3	$-5, -5$	$-5, -5$	$5, 0$	$0, 0$
	a_4	$-5, -5$	$-5, -5$	$0, 0$	$0, 5$

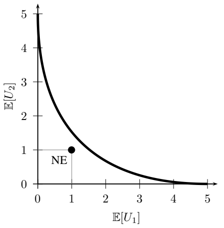

Figure 2: 2–agent game (left) and its Pareto frontier (right).

Proposition 6 *There are NEs that do not satisfy constraints (16)–(20).*

Proof. Consider the game in Fig. 3. We show that KKT conditions are not satisfied at the NE $(\mathsf{a}_2, \mathsf{a}_4)$. KKT conditions at $x_1(\mathsf{a}_2) = x_2(\mathsf{a}_4) = 1$ are: $\mu_1(\mathsf{a}_2) = 0$, $\mu_2(\mathsf{a}_4) = 0$, $5\lambda_2 + \mu_1(\mathsf{a}_1) = \nu_1$, $\lambda_1 + \lambda_2 = \nu_1$, $5\lambda_1 + \mu_2(\mathsf{a}_3) = \nu_2$, $\lambda_1 + \lambda_2 = \nu_2$. By straightforward mathematics, we obtain

$$3\lambda_1 + 3\lambda_2 = -\mu_1(\mathsf{a}_1) - \mu_2(\mathsf{a}_3)$$

Given that $\lambda_i, \mu_i(a_i) \geq 0$ and $\sum_{i \in N} \lambda = 1$, the above equality is unsatisfiable and therefore KKT conditions are not satisfied. ☐

		agent 2	
		a_3	a_4
agent 1	a_1	$3, 3$	$0, 5$
	a_2	$5, 0$	$1, 1$

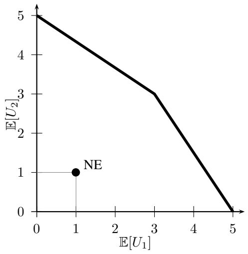

Figure 3: Example game (prisoner's dilemma) without any SNE (left) and Pareto frontier (right).

NEs on the correlated Pareto frontier

In this subsection, we present our third formulation approach. In order to provide sufficient conditions, we relax the constraints of program (5)–(8), allowing the agents belonging to coalition C to play correlated strategies. Correlated strategies include all the mixed strategies. Thus, if a strategy profile of the members of coalition C is the best w.r.t. all their correlated strategies, then it is the best also w.r.t. all the mixed strategies. However, requiring optimality w.r.t. correlated strategies, we may discard solutions that are optimal w.r.t. mixed strategies.

At first, we reformulate program (5)–(8) when the members of coalition C can play correlated strategies:

objective (5)

$$x_C(a) \geq 0 \qquad \forall a \in A_C \tag{21}$$

$$\sum_{a\in A_C} x_C(a) = 1 \tag{22}$$

Given that the above optimization problem is convex (i.e., linear in $\mathbf{x}_C$), we have that, if a solution $\mathbf{x}_C$ is optimal for program (5), (21), (22), then there is a vector of multipliers $\lambda_i \geq 0$ in which at least one multiplier is strictly positive such that $\mathbf{x}_C$ is an optimal solution of the following problem:

constraints (21), (22)

$$\max_{\mathbf{x}_C} \sum_{i\in C} \lambda_{i,C} \cdot \sum_{(a_C,a_{-C})\in A} U_i(a_C,a_{-C}) \cdot \\ \cdot x_C(a_C) \cdot \prod_{j\notin C} x_j(a_j) \tag{23}$$

We derive the dual problem of problem (21), (22), (23). It is:

$$\min_{v_C} v_C \tag{24}$$

$$v_C \geq \sum_{i\in C} \lambda_{i,C} \cdot \sum_{a_{-C}\in A_{-C}} U_i(a_C,a_{-C}) \cdot \\ \cdot \prod_{j\notin C} x_j(a_j) \;\; \forall a_c \in A_c \tag{25}$$

where v_C is the dual variable of constraint (22). Given that the primal problem is convex, strong duality holds and we can apply the complementary slackness theorem, obtaining the following feasibility problem:

constraints (21), (22), (25)

$$x_C(a_c) \cdot \Bigg(v_C - \sum_{i\in C} \lambda_{i,C} \cdot \sum_{a_{-C}\in A_{-C}} U_i(a_C,a_{-C}) \cdot \\ \cdot \prod_{j\notin C} x_j(a_j) \Bigg) = 0 \;\; \forall a_c \in A_c \tag{26}$$

The above constraints are sufficient conditions for a strategy profile $\mathbf{x}_C$ to be Pareto efficient once the strategies of agents outside C are fixed. Now, we can leverage the above results to produce a nonlinear program for SNE:

constraints (1), (2), (3), (4)

constraints (6), (25), (26) $\forall C \in \mathcal{C}$

$$\lambda_{i,C} \geq 0 \qquad \forall i \in C, C \in \mathcal{C} \tag{27}$$

$$\sum_{i\in C} \lambda_{i,C} = 1 \qquad \forall C \in \mathcal{C} \tag{28}$$

where constraints (1)–(4) assure that $\mathbf{x}$ is an NE; constraints (6), (25), (26) $\forall C \in \mathcal{C}$ and constraints (27), (28) assure that, for every C, there are some well–defined multipliers $\lambda_{i,C}$ such that $\mathbf{x}_C$ is optimal (among all the correlated strategies) and therefore $\mathbf{x}_C$ is Pareto efficient. We can state the following theorem, whose proof easily follows from above.

Theorem 7 *The above constraints are sufficient conditions for a strategy profile to be an SNE.*

However, the above conditions are not necessary:

Theorem 8 *The above constraints are not necessary conditions for a strategy profile to be an SNE.*

Proof. Consider Fig. 4, where ρ is arbitrarily small. The game has one SNE, i.e., $(\mathsf{a}_3, \mathsf{a}_6)$, but this SNE does not satisfy the above constraints. Indeed, the agents' utilities at $(\mathsf{a}_3, \mathsf{a}_6)$, i.e., $(2, 2)$, are not on the correlated–strategy Pareto frontier, i.e., the dashed line connecting (5,0) to (0,5). Thus, the above nonlinear mathematical program is infeasible. ☐

agent 2

agent 1

	a_4	a_5	a_6
a_1	$5,0$	$0,0$	$0,\rho$
a_2	$0,0$	$0,5$	$0,\rho$
a_3	$\rho,0$	$\rho,0$	$2,2$

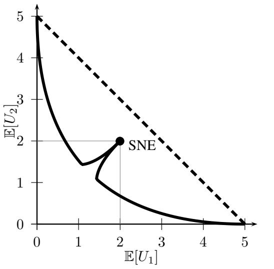

Figure 4: 2–agent game (left) and its Pareto frontier (right).

Relationships between the solutions

We now study the relationships between the solutions found by the formulations provided in the previous subsections. Call *NEPMDs* the NEs that are resilient to pure multilateral deviations, *NEKKTs* the NEs that satisfy the KKT conditions, and *corrSNEs* the SNEs that are Pareto efficient w.r.t. the correlated strategies.

Proposition 9 *NEKKTs $\not\subseteq$ NEPMDs and NEPMDs $\not\subseteq$ NEKKTs.*

Proof. We prove this by a pair of counterexamples. The NE $(\frac{1}{3}\mathsf{a}_1 + \frac{2}{3}\mathsf{a}_2, \frac{3}{4}\mathsf{a}_3 + \frac{1}{4}\mathsf{a}_4)$ of the game in Fig. 5 is an NEPMD, while it is not an NEKKT, and therefore NEKKTs do not constitute a subset of NEPMDs; the NE $(\mathsf{a}_3, \mathsf{a}_6)$ of the game in Fig. 6 is an NEKKT, while it is not an NEPMD, and therefore NEPMDs does not constitute a subset of NEKKTs. ☐

agent 2

agent 1

	a_3	a_4
a_1	$1,0$	$0,2$
a_2	$0,1$	$3,0$

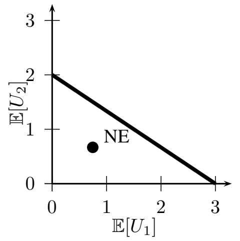

Figure 5: 2–agent game (left) and its Pareto frontier (right).

Proposition 10 *SNEs $\subset$ (NEKKTs $\cap$ NEPMDs).*

Proof. The NE $(\frac{1}{2}a_1 + \frac{1}{2}a_2, \frac{1}{2}a_5 + \frac{1}{2}a_6)$ of Fig. 2 is an NEPMD and an NEKKT, but it is not an SNE. ☐

So, the relationships between the solutions are:

		agent 2		
		a_4	a_5	a_6
agent 1	a_1	$5, 2$	$0, 0$	$0, 0$
	a_2	$0, 0$	$2, 5$	$0, 0$
	a_3	$0, 0$	$0, 0$	$1, 1$

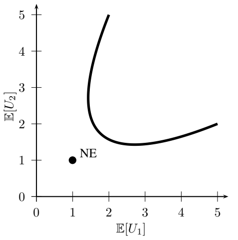

Figure 6: 2–agent game (left) and its Pareto frontier (right).

$$\text{corrSNEs} \subset \text{SNEs} \subset (\text{NEKKTs} \cap \text{NEPMDs}) \begin{array}{c} \subset \text{NEPMDs} \subset \\ \subset \text{NEKKTs} \subset \end{array} \text{NEs}$$

Algorithm for finding an SNE in games with more than two agents

Recently a tree search algorithm was presented for finding an SNE in 2–agent games (Gatti, Rocco, and Sandholm 2013). We extend it to multiple agents. For simplicity, we present the new tree search algorithm for three agents, but the generalization to more than three is immediate.

The 2–agent algorithm works as follows. A state s is a subspace of the agents' utility space defined as $s = [U_1^{\min}, U_1^{\max}] \times [U_2^{\min}, U_2^{\max}]$. The algorithm is first called with s being the entire space. At each call of the algorithm, the algorithm calls an NE–finding oracle to find an NE (or to state that none exists) in the subspace s. If there is an NE, a verification algorithm is then called to check whether the NE is an SNE by checking whether there is a solution with utility $(\overline{U}_1, \overline{U}_2)$ that Pareto dominates the NE. The oracle that was adopted is MIP Nash (Sandholm, Gilpin, and Conitzer 2005). The verification algorithm is described in (Gatti, Rocco, and Sandholm 2013) and its complexity is polynomial. If there is a Pareto dominant solution $(\overline{U}_1, \overline{U}_2)$—so the NE is not an SNE—the algorithm generates two states $s_1 = [U_1^{\min}, U_1^{\max}] \times [\overline{U}_2, U_2^{\max}]$ and $s_2 = [\overline{U}_1, U_1^{\max}] \times [U_2^{\min}, \overline{U}_2]$, and calls itself with these states recursively.

We extend the algorithm to the 3–agent case as follows.

First, we need to generalize the tree search framework. States are defined as $s = [U_1^{\min}, U_1^{\max}] \times [U_2^{\min}, U_2^{\max}] \times [U_3^{\min}, U_3^{\max}]$. In addition, there are now special states s^* defined as $[U_i^{\min}, U_i^{\max}] \times [U_j^{\min}, U_j^{\max}] \cup \{\mathbf{x}_k\}$, where the strategy of agent k is fixed. For states s, an SNE–candidate–finding oracle for 3–agent games is called. For s^*, an SNE–candidate–finding oracle for 2–agent games is called. At each iteration the algorithm randomly chooses a state s or s^*. Given an NE $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$ in $s = [U_1^{\min}, U_1^{\max}] \times [U_2^{\min}, U_2^{\max}] \times [U_3^{\min}, U_3^{\max}]$, if there is a Pareto dominant solution $(\overline{U}_1, \overline{U}_2, \overline{U}_3)$ for the grand coalition, we generate

- $s_1 = [\overline{U}_1, U_1^{\max}] \times [U_2^{\min}, U_2^{\max}] \times [U_3^{\min}, U_3^{\max}]$,
- $s_2 = [U_1^{\min}, \overline{U}_1] \times [\overline{U}_2, U_2^{\max}] \times [U_3^{\min}, U_3^{\max}]$,
- $s_3 = [U_1^{\min}, \overline{U}_1] \times [U_2^{\min}, \overline{U}_2] \times [\overline{U}_3, U_3^{\max}]$;

if there is a Pareto dominant solution $(\overline{U}_i, \overline{U}_j)$ for the coalition $C = \{i, j\}$, we generate

- $s_1^* = [U_i^{\min}, U_i^{\max}] \times [\overline{U}_j, U_j^{\max}] \cup \{\mathbf{x}_k\}$,
- $s_2^* = [\overline{U}_i, U_i^{\max}] \times [U_j^{\min}, \overline{U}_j] \cup \{\mathbf{x}_k\}$,

where $k \in N \setminus C$. If s^* turns out to be Pareto dominated, we generate the same states generated with 2–agent games, keeping fix the strategy of $i \in -C$. The algorithm is called recursively on each of the newly–generated states. Finally, we need to check dominance considering all the three agents.

Second, for the oracle, we need a different method than in the 2–agent case because MIP Nash (Sandholm, Gilpin, and Conitzer 2005) is based on integer linear programming, and with more than two agents the NE–finding program is nonlinear. For the oracle, we adopt (from earlier in this paper) formulations for NEs, NEPMDs (in place of integrity constraints $r \in \{0, 1\}$, we use constraints $r \cdot (1 - r) = 0$ and $r \geq 0$), NEKKTs, NEPMDs∩NEKKTs. As we proved, these are necessary but not sufficient conditions for SNE. Once such a candidate solution is found, we check whether it really is an SNE using the same algorithm as in the prior 2–agent algorithm, and that verification algorithm is still polynomial. If the oracle finds just an NE at that tree search node although an SNE exists at that node, that is no problem because the SNE will be encountered later in the tree search. On the other hand, if the oracle fails to find any NE at that tree search node—e.g., because the nonlinear mathematical program of the oracle is solved with an incomplete algorithm—although an SNE exists at that tree search node, the overall algorithm can become incomplete.

Experiments

We implemented the tree search algorithm in the C programming language. For the oracle formulations therein, we used AMPL (Fourer, Gay, and Kernighan 1990) as the modeling language and SNOPT (Stanford Business Software Inc. 2012) to solve them. The experiments were conducted on an Intel 2.20GHz processor with Linux kernel 2.6.32. We ran on 20 instances of class RandomGames[1] of GAMUT (Nudelman et al. 2004) for each setting of the parameters: 10, 20, and 25 actions for 2–agent games, $2, \ldots, 10$ actions for 3–agent games.

SNOPT uses sparse sequential quadratic programming and thus is not complete.[2] To make the oracle more complete, we used uniform random restarts over the strategy space (Onn and Weissman 2011) and called SNOPT on each restart. In Tab. 1 we report the percentage of runs where SNOPT finds an NE in 2–agent games: it shows that 50 random restarts are needed to have satisfactory reliability. One run of SNOPT on the formulations NEMDP, NEKKT, and corrSNE required compute time similar to that reported in

---

[1] We report data for RandomGames because it is the most representative class among the GAMUT classes.

[2] We also evaluated global optimization solver Couenne (http://www.coin-or.org/Couenne/). Turns out it was applicable only to instances with less than 5 actions in 2–agent games, since it required a long time (over an hour) on larger instances.

the table. The compute time per restart is relatively small (few seconds), but, due to the large number of restarts, it is much larger (by about two orders of magnitude) than the compute time needed by MIP Nash, which takes less than one second and does not require restarts (Sandholm, Gilpin, and Conitzer 2005).

		average time per restart	restarts				
			10	20	30	40	50
actions	10	2.1 s	50%	80%	90%	95%	98%
per	20	3.2 s	40%	65%	73%	85%	96%
agent	25	6.3 s	30%	50%	63%	77%	94%

Table 1: SNOPT performance in finding an NE.

With 50 random restarts of SNOPT in each oracle call, we evaluated our approaches of finding an SNE (or stating that none exists) on 3–agent games, Tab. 2. The average time per restart is similar for all the formulations (NEMDP, NEKKT, and corrSNE) and, due to limited space, we report only the average across the formulations. The average number of tree search nodes (i.e., calls to the oracle) per formulation shows that our new necessary formulations for SNE significantly reduce the number of nodes compared to just using an NE–finding oracle. Finally, corrSNE returns the correct solution with low percentage.

actions per agent	av. time per restart	tree search nodes (i.e., calls to the oracle)				success corrSNE
		NE	NEPMD	NEKKT	NEPMD∩NEKKT	
5	6.3 s	3.0	1.0	1.3	1.0	45%
6	8.6 s	3.6	1.2	1.6	1.0	25%
7	10.1 s	4.1	1.1	1.5	1.1	20%
8	15.6 s	4.7	1.0	1.8	1.0	15%
9	19.4 s	6.4	1.1	1.7	1.1	10%
10	25.8 s	5.6	1.1	1.8	1.1	10%

Table 2: Performance to find an SNE (or state that none exists) in 3–agent games.

In 2–agent games, the reduction in the number of tree search nodes is similar to the one in Tab. 2 (less than one order of magnitude), but, in this case, it is not enough to make our new nonlinear formulations faster than MIP Nash. This is because, as said above, the new oracles requires a much longer time due to random restarts.

Conclusions and future research

In this paper, we presented the first general–purpose algorithms for strong Nash equilibrium (SNE) finding in games with more than two agents. We first derived a nonlinear program for finding a Nash equilibrium (NE) that is resilient to pure–strategy coalitional deviations, and showed that it is a necessary condition for SNE but not sufficient. Second, we derived a nonlinear program to find NEs that satisfy Karush–Kuhn–Tucker conditions and showed that it is necessary for SNE but not sufficient. Third, we derived a nonlinear program to find NEs that are Pareto efficient for each coalition with respect to coalition correlated strategies, and showed that it is sufficient for SNE but not necessary. The problem whether there is a necessary and sufficient set of equilibrium constraints in mathematical programming fashion is left open. Then, we developed a tree search algorithm for SNE finding, and leveraged our necessary conditions to obtain better oracles for use at the search tree nodes.

Experiments showed the viability of the approach. Using the new necessary conditions in the oracle significantly reduces search tree size compared to using NE conditions alone. Also, the sufficient conditions often yield an SNE, but as the game size increases, this becomes less likely.

This work underlined that nonlinear mathematical programming tools—albeit necessary because the SNE problem with more than two agents is nonlinear—have drawbacks: they require random restarts and may cause incompleteness. In future work, we plan to evaluate in depth the performance of both non–linear programming solvers and mathematical programming with equilibrium constraints (Luo, Pang, and Ralph 1996) solvers. We also plan to study approximation algorithms for SNE. Finally, we plan to study SNE finding in compactly representable games that do not require nonlinear programming, such as polymatrix games.

Acknowledgments

N. Gatti was supported by MIUR under grant 2009BZM837. T. Sandholm was supported by NSF under grants IIS–0964579 and CCF–1101668.

References

Aumann, R. 1960. Acceptable points in games of perfect information. *PAC J MATH* 10:381–417.

Chen, X.; Deng, X.; and Teng, S.-H. 2009. Settling the complexity of computing two–player Nash equilibria. *J ACM* 56(3):14:1–14:57.

Conitzer, V., and Sandholm, T. 2008. New complexity results about Nash equilibria. *GAME ECON BEHAV* 63(2):621–641.

Daskalakis, C.; Goldberg, P.; and Papadimitriou, C. 2006. The complexity of computing a Nash equilibrium. In *STOC*, 71–78.

Epstein, A.; Feldman, M.; and Mansour, Y. 2007. Strong equilibrium in cost sharing connection games. In *ACM EC*, 84–92.

Fourer, R.; Gay, D.; and Kernighan, B. 1990. A modeling language for mathematical programming. *Management Science* 36(5):519–554.

Gatti, N.; Patrini, G.; Rocco, M.; and Sandholm, T. 2012. Combining local search techniques and path following for bimatrix games. In *UAI*, 286–295.

Gatti, N.; Rocco, M.; and Sandholm, T. 2013. On the verification and computation of strong Nash equilibrium. In *AAMAS*.

Gourvès, L., and Monnot, J. 2009. On strong equilibria in the max cut game. In *WINE*, 608–615.

Hayrapetyan, A.; Tardos, E.; and Wexler, T. 2006. The effect of collusion in congestion games. In *STOC*, 89–98.

Hoefer, M., and Skopalik, A. 2010. On the complexity of Pareto–optimal Nash and strong equilibria. In *SAGT*, 312–322.

Holzman, R., and Law-Yone, N. 1997. Strong equilibrium in congestion games. *GAME ECON BEHAV* 21:85–101.

Lemke, C., and Howson, J. 1964. Equilibrium points of bimatrix games. *SIAM J APPL MATH* 12(2):413–423.

Luo, Z.-Q.; Pang, J.-S.; and Ralph, D. 1996. *Mathematical Programs with Equilibrium Constraints*. Cambridge University Press.

Miettinen, K. 1999. *Multiobjective Optimization*. Kluwer.

Nessah, R., and Tian, G. 2012. On the existence of strong Nash equilibria. *IESEG School of Management, Working Paper*.

Nudelman, E.; Wortman, J.; Leyton-Brown, K.; and Shoham, Y. 2004. Run the GAMUT: A comprehensive approach to evaluating game–theoretic algorithms. In *AAMAS*, 880–887.

Onn, S., and Weissman, I. 2011. Generating uniform random vectors over a simplex with implications to the volume of a certain polytope and to multivariate extremes. *Annals of Operations Research* 189(1):331–342.

Porter, R.; Nudelman, E.; and Shoham, Y. 2009. Simple search methods for finding a Nash equilibrium. *GAME ECON BEHAV* 63:642–662.

Rozenfeld, O., and Tennenholtz, M. 2006. Strong and correlated strong equilibria in monotone congestion games. In *WINE*, 74–86.

Sandholm, T.; Gilpin, A.; and Conitzer, V. 2005. Mixed–integer programming methods for finding Nash equilibria. In *AAAI*, 495–501.

Shoham, Y., and Leyton-Brown, K. 2008. *Multiagent Systems: Algorithmic, Game Theoretic and Logical Foundations*. Cambridge University Press.

Stanford Business Software Inc. 2012. http://www.sbsi-sol-optimize.com/.

Thompson, D. R. M.; Leung, S.; and Leyton-Brown, K. 2011. Computing Nash equilibria of action–graph games via support enumeration. In *WINE*, 338–350.