Distributed constrained search by selfish agents for Efficient equilibria

Vadim Levit and Amnon Meisels
Dept. of Computer Science
Ben-Gurion University

Abstract

Search for stable solutions in games is a hard problem that includes two families of constraints. The global stability constraint and multiple soft constraints that express preferences for socially, or otherwise, preferred solutions. To find stable solutions (e.g., pure Nash equilibria - PNEs) of high efficiency, the multiple agents of the game perform a distributed search on an asymmetric distributed constraints optimization problem (ADCOP). Approximate (local) distributed search on ADCOPs does not necessarily guarantee convergence to an outcome that satisfies the stability constraints, as well as optimizes the soft constraints. The present paper proposes a distributed search algorithm that uses transfer of funds among selfish agents. The final outcome of the algorithm can be stabilized by transfer of funds among the agents, where the transfer function is contracted among the agents during search. It is shown that the proposed algorithm - Iterative Nash Efficiency enhancement Algorithm (INEA) - guarantees improved efficiency for any initial outcome.

The proposed distributed search algorithm can be looked at as an extension to best response dynamics, that uses transfer functions to guarantee convergence and enforce stability in games. The best-response-like nature of INEA establishes its correct behavior for selfish agents in a multi-agents game environment. Most important, unlike best response, the proposed INEA converges to efficient and stable outcomes even in games that are not potential games.

Introduction

A common solution concept for games played by multiple agents is the Nash equilibrium, i.e, a stable state in which no participant can gain by a unilateral change of strategy (cf. [1]). Finding a PNE in a general game and in particular an efficient one forms a constrained search problem, where stability is a global hard constraint and efficiency is composed of soft constraints for which a good solution (e.g., of higher social welfare) is sought (cf. [2]). The problem of finding a Nash equilibrium has been shown to be computationally intensive [1, 3] and heuristic search algorithms for this constrained problem have been recently proposed [4, 5]. Additionally, it is known that equilibria states are not necessarily efficient with respect to a global objective [6, 3] or even with respect to the personal payoffs of the agents [7].

The distributed nature of games, played by multiple agents, requires distributed search methods to find solutions. Several distributed complete search algorithms for finding stable states in games were proposed [8, 9, 10]. However, for games among a large number of agents the problem of finding a PNE becomes computationally intractable [1] and one needs to rely on approximate search which can find a PNE in reasonable time (cf. [2]). Complete distributed search assumes cooperation among the participating agents [11], but this assumpltion is not always acceptable for selfish agents. One simple way of avoiding the assumption of cooperation among the searching agents is to apply specially designed local search. However, the major problem of approximate (local) search is its ability to converge, as well as guarantee that the resulting solution will be stable.

On a different theme, approximate search methods were shown to converge to a stable state in *Potential games* [12], where best-response dynamics converge to a PNE. When an approximate search method like best response converges to a stable solution, the efficiency of the solution can be arbitrarily low. In contrast, it is well known that outcomes of high efficiency are not necessarily stable (e.g. "the prisoners' dilemma"). This tradeoff between efficiency and stability motivates the use of an incentive mechanism to promote stability in efficient states. Jackson and Wilkie [13] have studied the efficiency of equilibria that are achieved by using *side payments*. The side payments mechanism enables to incentivize unsatisfied agents to agree on some preferred outcomes. Side payments allow transfers of funds (payoffs) between agents, such that agents who gain from some outcome may want to pay others that are unsatisfied by it.

The present paper proposes an incomplete decentralized search algorithm for selfish agents, to find a stable state of higher efficiency. One can view the proposed Iterative Nash Efficiency-enhancement Algorithm (INEA) as an enhancement to the best-response dynamics (cf. [12]). The innovation of INEA is that it enables participating agents to sacrifice part of their payoff at a certain outcome in order to convince other agents to play a certain strategy. In this sense, the inter-agent transfers of funds are used in order to ensure the convergence of the algorithm to outcomes which are both stable and efficient. The outcome resulting from the run of the INEA algorithm is ensured to be at least as efficient as the original outcome. Most importantly, it is guaranteed that the resulting outcome can be transformed into a stable state by the use of the transfers of funds that were contracted by the agents during the run of the algorithm.

The proposed INEA algorithm uses a fixed order of all the agents. Each agent in its turn exchanges messages with its neighbors and decides on its selected strategy. The messages may include proposed transfers of funds among the interacting agents. The game is assumed to have interactions among a limited number of agents at each step, which is equivalent to assuming some underlying graph or social network. This compact general structure for multi-agents games has been termed graphical games in the past [14] and they are known to have a compact representation. For clarity, the term multi-agents games (MAGs) will be used through the rest of the paper. Since the distributed search algorithm proposed by the present paper uses strategic reasoning by the agents that compute their transfers of payoffs, transfer functions can be thought of as *side payments* [13]. Side payments that are contracted by the agents during the run of the algorithm, to be transferred for its final outcome [13, 15, 16].

An extensive experimental evaluation of the INEA algorithm demonstrates two main results. First, that for multi-agents games that are also potential games, the proposed algorithm produces stable states that are more efficient than best response and in fact that the resulting stable states are close to optimal. Second, that for randomly generated MAGs the fairness of the efficient stable states that are produced by the INEA algorithm is highly affected by the nature of the transfer function used.

Section **Preliminaries** introduces the needed concepts of games and their stable states, pure Nash equilibria (PNEs). It emphasizes the equivalence of multi-agents games with distributed constraints search. **Preliminaries** includes the definitions of transfers of funds among agents as well as the definition of outcomes which can be stabilized by the use of side payments. The proposed Iterative Nash Efficiency enhancement Algorithm is described in detail in section **Finding Efficient Equilibria** together with proofs for the main guarantees of the algorithm. A detailed example of the run of the proposed algorithm on an example problem is also provided. The following section presents the experimental evaluation of INEA. First, the INEA algorithm is compared to best-response on problems that are potential games. Next, two forms of transfer functions are compared on randomly generated problems that are not necessarily potential games.

Preliminaries

Multi-agents games

Multi-agents games (MAGs) consist of a set of agents $A = \{1, \ldots, n\}$, where each agent i has a set of strategies V_i from which it can select a strategy $v_i \in V_i$ to play. In the terminology of distributed constraints problems, strategies are *assignments* [8] and the games among agents are equivalent to asymmetric constraints [11, 9]. An *outcome* of the multi-agents game, $v = (v_1, \ldots, v_n) \in V_1 \times \ldots \times V_n$, is a collection of strategies (e.g., assignments), one for each agent. Each agent i interacts (plays) with some set of other agents, which we term N_i (e.g., its neighborhood in graphical games terms [14]) and can be a small subset of A. Agent i's payoff function is denoted $u_i : \times_{j \in N_i \cup i} V_j \rightarrow \mathbb{R}$, where N_i is usually limited to a small subset of A.

Multi-agents games that have agents interacting with a small numbers of "neighboring" agents are commonly termed graphical games and have a compact representation [14]. The agents can be thought of as if they are connected by some underlying graph (e.g., social network) $G = \{N, E\}$. Each vertex in N represents an agent and edges in E represent the interaction structure of the game. As stated above, game-like interactions are asymmetric constraints. Given an agent $i \in N$, the set of i's neighbors N_i are the agents whose actions impact i's payoff [17]. Note that the classical Normal-form game can be represented as a MAG having a complete underlying graph. The assumption of a small size of N_i is used in order to improve the run-time of the computation of the transfers of payoffs and to simplify the presentation of the proposed distributed search algorithm.

The *pure Nash Equilibrium (PNE)* is a central concept in game theory (cf. [1, 3]).

We say that an outcome is a PNE, if every agent does not prefer to change its strategy (assignmnt), given the strategies (assignments) of all other agents in this outcome. Formally, an outcome v is a PNE if the following holds:

$$\forall i \in N, \nexists v'_i \in V_i \text{ s.t., } u_i(v_i, v_{-i}) < u_i(v'_i, v_{-i}) \quad (1)$$

where v_{-i} is the standard notation for the combined strategies (assignments) of all agents except i.

Finding a PNE in the above description of a MAG is equivalent to solving an Asymmetric Constraints Optimization Problem (ADCOP) [11], where the agents' domains of values are the palettes of strategies of agents in the MAG and the constraints among neighbors of an ADCOP are represented by the values in the normal form game matrix (cf. [8, 16, 18]). Stability is represented by the above global constraint of the ADCOP. Once an ADCOP is constructed, the finding of an efficient PNE can be solved by an appropriate algorithm (cf. [19, 8]). The present paper focuses on local search algorithms that can be run by rational agents, emphasizing a distributed search protocol that bears similarities to the game theoretic best-response dynamics [12].

Transfer functions and Side payments

Transfer functions endow agents with the possibility of sacrificing part of their payoff in order to convince other agents to play a certain strategy. Given a multi-agents game, transfers of payoff are defined by a function $\tau : N \times N \times V_1 \times \ldots \times V_n \rightarrow \mathbb{R}^+$, where $\tau_{i,j}(v)$ denotes the payment being transferred from agent i to agent j in outcome v. In order to take transfer functions into consideration while deciding on the action to take, an agent's utility must reflect the change in its payoff. The *net loss* that is incurred on agent i at outcome v when using the transfer function τ is defined by equation 2.

$$\tau_i(v) := \sum_{j \in N_i} (\tau_{i,j}(v) - \tau_{j,i}(v)) \quad (2)$$

We restrict our attention to transfer functions such that if $j \notin N_i$ then $\forall v : \tau_{i,j}(v) = 0$. In words, side payments take place between neighbors in the game. Note again that neighbors are connected by a two-players game which is an asymmetric binary constraint. Given the definition of net loss (Equation 2), one can update the definition of the utility that agent i obtains from outcome v as follows:

$$u_i^\tau(v) := u_i(v) - \tau_i(v) \quad (3)$$

Transfer functions have been termed *Side payments* by Jackson and Wilkie [13] when their values are decided upon by the agents that strategically reason about their values. Side payments enable games to be transformed from the inside and are defined as transfers of payoffs between agents, where each agent may pay (or receive payment from) each one of its neighbors. Since the distributed search algorithm proposed by the present paper uses strategic reasoning by the agents that compute their offered and accepted transfers of payoffs, we will use the terms transfer functions and side payments interchangeably.

The use of transfer functions in the present study is mainly motivated by the goal to obtain a stable state (PNE) with certain properties. Let us start by differentiating between outcomes that can be transformed into a PNE and those that cannot.

Definition 1. *An outcome v in a multi-agents game G is* side payments enforceable (SPE) *if there exists a transfer function τ, such that:*

$$\forall i \in N,\ \nexists v'_i \in V_i \text{ s.t., } u^{\tau}_i(v_i, v_{-i}) < u^{\tau}_i(v'_i, v_{-i})$$

Definition 1 ensures that the updated agents utilities (according to the transfer function τ) result in a PNE at outcome v, satisfying Equation 1.

The intuitive argument for how transfers of payoffs have the ability to obtain stable outcomes goes as follows: An agent can offer a neighboring agent compensation which is a function of the second agent's action, such that the compensation effectively reflects any utility loss that the second agent's action incurs on the first agent. Take for example the following case. The utility loss of agent i is x if agent j takes action v'_j rather than its current action v_j, while the benefit of agent j of taking action v'_j rather than v_j is only y where $x > y$. Agent i in this example can offer to agent j a monetary compensation of z, where $x \geq z \geq y$, if agent j will not deviate from its current strategy v_j. Note that any z such that $x \geq z \geq y$ will provide a sufficient incentive for agent j.

Finding Efficient Equilibria

The proposed distributed search algorithm, Iterative Nash Efficiency enhancement Algorithm (INEA), relies on inter-agent transfers of payoffs which is a key element in the proposed procedure. The INEA search algorithm is iterative, where each iteration goes over all agents in a predefined fixed order. Each agent in its turn proposes its selected strategy by sending messages to its neighbors and receiving from them response messages if they wish to transfer funds to the proposing agent in order to convince it to avoid taking the proposed strategy. The algorithm is proven to converge to an SPE global state with improved (or at least the same) global efficiency. In addition, this distributed method may be complemented by existing heuristics that find initial outcomes of high efficiency. The resulting combination (which is not pursued in the present study) has the potential to produce in its end result PNEs of even higher efficiency.

The INEA algorithm

The INEA algorithm is composed of two consecutive stages. Starting from an initial outcome v, the main stage iterates over all agents (in a predefined fixed order) until it converges to an outcome v^*. The efficiency of outcome v^* is proven to be at least as high as the efficiency of the initial outcome v. Outcome v^* is not necessarily a stable one, but is guaranteed to be side payments enforceable (SPE). A set of transfers that can be used as side payments among the agents to enforce a PNE is computed by the agents during the iterations and is guaranteed to not worsen the utility of all agents. It therefore can be thought of as a binding contract among the agents during the iterative search, computed by the neighboring (e.g., interacting) agents during each step. In that sense one can call them side payments [13] and think of the proposed INEA algorithm as an iterative method for searching for both an efficient outcome and the side payments that can guarantee its stability. Each iteration ends with an outcome and side payments that were computed during the iteration. The final outcome is reached when for a complete iteration over all agents no agent wishes to change its strategy. Exchanging messages with neighbors on the graph of the game makes the INEA algorithm a distributed local search algorithm (cf. [20, 21, 22]). It is important to understand that the major problem of standard local (approximate) search in DCOPs and ADCOPs is its ability to converge, as well as guarantee that the resulting solution will be stable [20, 21].

In the second stage of the run of the proposed INEA algorithm the transfers of payoffs among the agents are applied according to the computed contracts so that the outcome v^* is transformed into a stable state (i.e., a PNE). One can think of the proposed INEA algorithm as an iterative computation of the final outcome and the side payments that guarantee its stability, where all computations are performed by the agents in a distributed manner.

Main Stage

One can view the INEA algorithm (Algorithm 1) as an extended version of the best-response dynamics, extended with payoff transfers (contracts) among selfish agents. During the iterative run of the algorithm the participating agents decide on their transfer contracts to the other agents as well as the outcome which can be stabilized by applying these contracts.

All agents perform the main stage in a predefined fixed order, and each agent in turn executes the function **onStrategySelect()**. We will term the agent performing function **onStrategySelect()** the *current agent*. During the execution of Algorithm 1 the agents agree on contracts with their neighbors which are the incentive to the agents to remain with their choices (i.e., not change their strategy to the proposed one). Therefore, in order to decide regarding the choice of action, the current agent must take into consideration not only its own utility but also the compensation (contracts) from its neighbors as can be observed from lines 1-4 of Algorithm 1. Contracts are dissolved only due to the decision of an agent to change its strategy. Consequently, if an agent notifies its neighbors regarding the desire to change its strategy (line 5), all contracts concluded between its neighbors and itself are considered to be nullified. As a response to the message from the current agent, proposing to change its strategy, each neighbor decides upon the payoff transfer it wishes to sacrifice in order to convince the current agent to remain with its choice. These choices are sent back to the current agent and considered as new contracts only in case the agent remains with its choice. When the current agent receives responses from all of its neighbors, the agent will not choose the strategy that improves its utility only in the case that the agent's neighbors can secure for it a larger payoff (lines 7-8). This is a completely selfish (e.g., rational) decision when taking into consideration transfer functions.

Algorithm 1 defines the communication protocol among agents which enables them to decide about the transfers of payoff that can be used to stabilize the final outcome. Nevertheless, INEA does not explicitly specify how the decision about the exact mon-

Algorithm 1 Iterative Nash efficiency enhancement algorithm

onStrategySelect()

1: **let** $\mathbb{T} \leftarrow \sum_{j \in N_i} T_{j,i}$
2: **if** $\nexists v'_i$ s.t. $u_i(v'_i, v_{-i}) > u_i(v_i, v_{-i}) + \mathbb{T}$ **then**
3:     **return**
4: **select** v'_i s.t. $u_i(v'_i, v_{-i}) > u_i(v_i, v_{-i}) + \mathbb{T}$
5: **send**(**choice**, v'_i) to all $j \in N_i$
6: **let** $\mathbb{T}'$ be the sum of replies from all $j \in N_i$
7: **if** $u_i(v'_i, v_{-i}) > u_i(v_i, v_{-i}) + \mathbb{T}'$ **then**
8:     **update** $v \leftarrow (v'_i, v_{-i})$

when received(choice, v'_j)

9: **chose** $T_{i,j}$
10: **reply** $T_{i,j}$

etary transfer is made (line 9). Such a decision can have a great impact on the ability of the contracts to transform the outcome into a stable one. We will restrict our attention only to *admissible contracts* (defined below).

Definition 2. *Given an outcome v and a beneficial deviation v'_i of the current agent i, the contracts $T_{j,i}$ are* admissible *if the following conditions hold:*

1. $\forall j \in N_i, 0 \leq T_{j,i} \leq u_j(v'_i, v_{-i}) - u_j(v_i, v_{-i})$

2. if $\sum\limits_{j \in N_i, u_j(v_i, v_{-i}) > u_j(v'_i, v_{-i})} u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i}) \geq u_i(v'_i, v_{-i}) - u_i(v_i, v_{-i})$
then $\sum\limits_{j \in N_i} T_{j,i} \geq u_i(v'_i, v_{-i}) - u_i(v_i, v_{-i})$

Condition 1 restricts the proposed compensations to be "acceptable" by rational agents. It will be greater than zero only in the case that i's proposed change of strategy will incur a negative utility change for the responding neighbor. Additionally, an agent will not propose a transfer of payoff that exceeds its loss of utility. Condition 2 ensures that if the sum of the negative utility changes to neighbors exceeds the benefit of the current agent from its proposed unilateral deviation, so should the compensations. In other words, the proposed side-payments reflect the loss to the neighbors (from the proposed change of strategy of the current agent) and is therefore large enough to compensate for them.

Note that the contracts among agents do not affect the social welfare of outcomes. Therefore, one can omit the payoff transfers (i.e., take into consideration only the "original" utilities of the agents) when computing the social welfare. However, it is necessary to take the payoff transfers into consideration when reasoning about the stability of an outcome.

Correctness

Proposition 1. *Every update of an agent's strategy, performed by following Algorithm 1, improves the social welfare of the outcome v.*

Proof. According to Algorithm 1 only a current agent i which can improve its payoff by deviating from strategy v_i to v_i' can update its strategy. Such an update occurs only in the case where the sum of transfers offered by the responses of i's neighbors is smaller than the improvement in i's utility resulting from the strategy change (lines 7-8).

Suppose by contradiction that the current agent i updated its strategy from v_i to v_i' (running Algorithm 1) which improves i's utility but decreases the global social welfare. Clearly, all transfers do not change the social welfare because they represent only movement of funds from one agent to another. Consequently, the change in social welfare for the above case can be defined irrespective of the transfers as follows:

$$\sum_{j \in N} u_j(v_i', v_{-i}) - \sum_{j \in N} u_j(v_i, v_{-i}) \tag{4}$$

and will affect only the utilities of agent i and its neighbors. Therefore, Equation 4 can be simplified to:

$$(u_i(v_i', v_{-i}) - u_i(v_i, v_{-i})) + \sum_{j \in N_i} (u_j(v_i', v_{-i}) - u_j(v_i, v_{-i})) \tag{5}$$

Since the social welfare was decreased by deviation from v_i to v_i', Equation 6 must hold.

$$\begin{aligned} u_i(v_i', v_{-i}) - u_i(v_i, v_{-i}) &< \sum_{j \in N_i} (u_j(v_i, v_{-i}) - u_j(v_i', v_{-i})) \\ &\leq \sum_{j \in N_i, u_j(v_i, v_{-i}) > u_j(v_i', v_{-i})} (u_j(v_i, v_{-i}) - u_j(v_i', v_{-i})) \end{aligned} \tag{6}$$

Since only admissible contracts are considered, condition 2 of the admissible contracts requires that $\sum_{j \in N_i} T_{j,i} \geq u_i(v_i', v_{-i}) - u_i(v_i, v_{-i})$. Consequently, $u_i(v_i', v_{-i}) \leq u_i(v_i, v_{-i}) + \sum_{j \in N_i} T_{j,i}$ holds. As a result, the decrement in social welfare results in a contradiction to the condition of line 7 of Algorithm 1 and cannot be true. □

Corollary 2. *Algorithm 1 converges in a finite number of steps.*

The correctness of Corollary 2 follows directly from Proposition 1. Since each update of an agent's strategy according to Algorithm 1 improves the Social welfare of the outcome v and the Social welfare is bounded, the proposed algorithm terminates in a finite number of steps.

Corollary 3. *For every outcome $v \in V$, the social welfare of v^* obtained by running Algorithm 1 is greater than or equal to the social welfare of v.*

Since Proposition 1 ensures that the social welfare of outcome v is non-decreasing (at each step), then so is the social welfare of the final outcome v^*, which is at least as high as the social welfare of v.

Observation 4. *A graphical game updated by admissible contracts is an ordinal potential game where the potential function is the social welfare.*

The correctness of Proposition 1 directly leads to Observation 4. Namely, if an agent can change its strategy so as to increase its utility then the social welfare will be improved. Note that the opposite direction does not necessarily hold, i.e., a strategy change which decreases the agent's payoff does not necessarily result in a decrease of the overall social welfare.

In order to get some intuition about the differences between best-response dynamics and Algorithm 1, consider the following example:

Example 1. *Three agents are connected by edges, each representing a two-players game (e.g, an asymmetric constraint) on the graph in Figure 1. Every agent has exactly two strategies, a or b. The utilities of agents are the sum of payoffs of the two-player games (constraints) in Figure 1.*

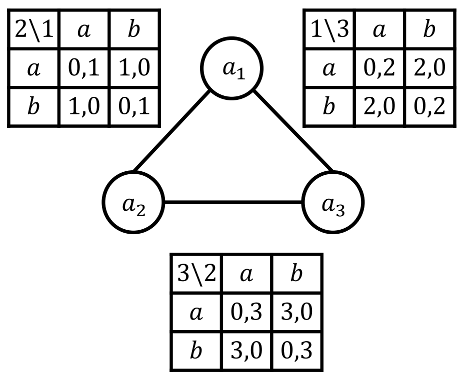

Figure 1: An example game with three agents

Consider an outcome $v = (v_1 = a, v_2 = a, v_3 = a)$, the payoff of agent a_1 is 1 in the game with a_2 and 0 in the game with a_3, leading to a total utility of 1 for agent a_1 in outcome v. The utility of agent a_2 is 0 in the game/constraint with a_1 and 3 against a_3, summing up to a total utility of 3 for a_2 in v. Finally, the utility of a_3 in outcome v is 2 since its payoff is 2 in the two-player game/constraint against a_1 and 0 in the game against a_2.

Let us start by following the run of the best response algorithm, where agents are ordered according to their indexes and the initial outcome is $v = (v_1 = a, v_2 = a, v_3 = a)$. To simplify the following of the agents reasoning in both algorithms, Table 1 presents the utilities of every agent in all 8 possible outcomes of the example game. Best response for agent a_1 is the unilateral deviation to strategy b which will increase its utility to 2. In the resulting outcome agent a_2 has no strategy which can improve its utility. Agent a_3 can change its strategy to b which will result in outcome $(v_1 = b, v_2 = a, v_3 = b)$. Now a_1 can improve its utility by changing to $v_1 = a$ which will be followed by agent a_2's move to strategy $v_2 = b$ and then agent a_3 will switch its strategy to a. Having arrived at the outcome $(v_1 = a, v_2 = b, v_3 = a)$, agent a_1 will benefit from

v_1	v_2	v_3	u_1	u_2	u_3
a	a	a	1	3	2
a	a	b	3	0	3
a	b	a	0	1	5
a	b	b	2	4	0
b	a	a	2	4	0
b	a	b	0	1	5
b	b	a	3	0	3
b	b	b	1	3	2

Table 1: The utilities of agents a_1, a_2 and a_3 in the example game.

deviating to $v_1 = b$. In the resulting outcome a_2 whose turn is next will change its strategy to a. It is easy to see that for our example game and outcome best-response dynamics has just closed a loop of states and does not converge.

Let us now follow the run of Algorithm 1 on the same game and outcome $v = (v_1 = a, v_2 = a, v_3 = a)$ and in the same order. The use of admissible contracts is assumed. Initially there are no transfers contracted and for all agents, $T_{i,j} = 0$.

As before, agent a_1 can improve its utility by deviating from a to b. Executing function ***onStrategySelect()****, it will send message $\langle$* ***choice****, $v_1 = b\rangle$ to its neighbors a_2 and a_3 (line 5 of Algorithm 1). The benefit to a_1 from this change of strategy is 1 (i.e., $u_1(v_1 = a, v_{-1}) = 1$ and $u_1(v_1 = b, v_{-1}) = 2$) but it will reduce the utility of agent a_3 from 2 to 0. For the admissibility of the contracts, agent a_3 will select a transfer of $1 \le T_{3,1} \le 2$ as incentive to agent a_1 to remain in its current strategy (line 7). The payoff of a_2 increases from the proposed change of a_1, so it does not respond at all. Since the outcome remains, a_2 has no strategy which improves its utility and it does not propose any (line 2 and 3). Agent a_3 will benefit from unilateral deviation from $v_3 = a$ to $v_3 = b$. However, a_2 will propose a sufficient incentive ($2 \le T_{2,3} \le 3$) for a_3 to remain with its current choice. This will suffice for a_3. No agent changed its strategy during this complete round and Algorithm 1 terminates with the stability enforceable outcome $v^* = (v_1 = a, v_2 = a, v_3 = a)$.*

It is easy to see that the final outcome of Example 1 is not a stable state. Agent a_1, for example, can change its strategy to b unilaterally and improve its gain from 1 to 2. However, this final state can be stabilized by using the transfers of funds that were contracted during the run of the algorithm as described above. In other words, it is guaranteed to be side payments enforceable. For example, agent a_3 can pay 1 to agent a_1 to secure its strategy and retain a stable state (a PNE).

Second Stage

When Algorithm 1 terminates there may be agents which have an incentive to unilaterally deviate from their strategy and improve their utility (similarly to Example 1). To stabilize outcome v^* the contracted transfers during the execution of Algorithm 1 need to be applied. The payoff transfers computed during the run of Algorithm 1 are as follows:

$$\tau_{j,i}(v) := \begin{cases} T_{j,i}, & \text{if } v_i = v_i^* \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Proposition 5. *For every outcome $v \in V$, the outcome v^* of Algorithm 1 is side payments enforceable.*

Proof. The correctness of Proposition 5 follows directly from the termination condition of Algorithm 1. When Algorithm 1 terminates, the only agents which must be incentivized in order to stabilize outcome v^* are those which can gain from unilateral deviation.

Formally, given outcome v an agent i has an incentive to unilaterally deviate if there exists a choice v_i' for it, such that $u_i(v_i', v_{-i}) > u_i(v_i, v_{-i})$. The termination of Algorithm 1 implies that for each agent i it holds that

$$\nexists v_i', u_i(v_i', v_{-i}) > u_i(v_i, v_{-i}) + \sum_{j \in N_i} T_{j,i}$$

which implies that

$$\nexists v_i', u_i(v_i', v_{-i}) - \sum_{j \in N_i} T_{i,j} > u_i(v_i, v_{-i}) + \sum_{j \in N_i} (T_{j,i} - T_{i,j})$$

and consequently, for each agent i the following holds:

$$\nexists v_i', u_i^\tau(v_i', v_{-i}) > u_i^\tau(v_i, v_{-i})$$

□

Observation 6. *An outcome v that maximizes social welfare is side payments enforceable.*

By Corollary 3, applying Algorithm 1 to an outcome v that maximizes social welfare will return the exact same outcome $v^* \equiv v$. By Proposition 5, this outcome is SPE.

The payoff transfer contracts defined by Algorithm 1 are sufficient in order to convince unsatisfied agents to stay in outcome v^* (Proposition 5). However, the INEA algorithm does not specify explicitly the actual exact transfer of payoff. It only assumes the admissibility of the contracts. Let us view two distinct admissible contract schemes which can be efficiently computed during the run of the INEA algorithm.

The first one does not make any assumption regarding the knowledge of agents about the utilities and the contracts of other agents. The agents also do not perform any negotiation with other agents in order to decide about the amount of payoffs.

Consider the maximal payoff transfer that agent j is willing to secure for its neighbor i in outcome v in order to prevent i's change of strategy to v_i':

$$\delta_{j,i}(v, v_i') := \begin{cases} u_j(v_i, v_{-i}) - u_j(v_i', v_{-i}), & \text{if } u_j(v_i, v_{-i}) > u_i(v_i', v_{-i}) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Using Equation 8 admissible contracts are:

$$T_{j,i} = \delta_{j,i}(v, v'_i) \tag{9}$$

These contracts are termed *maximal payoff contracts* and their computation does not need interaction among agents.

Observation 7. *Maximal payoff contracts are admissible.*

The admissibility of the maximal payoff contracts arises directly from the definition. Agent j will sacrifice a payoff transfer of exactly $T_{j,i} = u_j(v_i, v_{-i}) - u_j(v'_i, v_{-i})$ to agent i only if it equals the decrement in its utility due to the unilateral deviation of agent i from v_i to v'_i.

Another contracts scheme - *cooperative contracts* - assumes full knowledge of the game and cooperation among the agents in a neighborhood and the contracts are defined as follows:

$$T_{j,i} := \begin{cases} 0, \text{if } \sum\limits_{l \in N_i} \delta_{l,i}(v, v'_i) < u_i(v'_i, v_{-i}) - u_i(v_i, v_{-i}) \\ \delta_{j,i}(v, v'_i) \cdot \frac{u_i(v'_i, v_{-i}) - u_i(v_i, v_{-i})}{\sum\limits_{l \in N_i} \delta_{l,i}(v, v'_i)}, \text{otherwise} \end{cases} \tag{10}$$

The size of *cooperative transfers* is proportional to the loss in the agent's utility, which is analogous to Shapley's value [23].

Observation 8. *Cooperative contracts are admissible.*

This follows immediately from the definition in equation 10. Non-zero transfers are only proposed if their sum compensates the deviating agent and no neighboring agent transfers more than its loss.

Experimental Evaluation

To empirically demonstrate the effectiveness of the INEA algorithm (Algorithm 1), two families of experiments were run. One family compares the performance of the INEA algorithm to best-response dynamics. Both algorithms converge to a stable point – best-response to a PNE and Algorithm 1 to an SPE. The other family of experiments studies the behavior of different admissible transfer functions. All the games that are used in the experimental evaluation fall under the description of *graphical games* [14, 9], where interaction among players/agents is limited to a small fraction (i.e., a neighborhood) of the overall large number of players.

Problem Generation

The first set of experiments uses "best-shot" public goods games [24, 17], which were proven to be potential games [18]. In "best-shot" public goods games each agent chooses an action $v_i \in \{0, 1\}$. For example, the action might be buying a book or some other product that is easily lent from one agent to another. Taking the action 1 is costly and an agent prefers a neighbor $j \in N_i$ taking the action than having to take it (N_i is i's neighborhood) But, taking the action and paying the cost is better than having nobody take the action. More formally, agent i's utility at outcome v is

$$u_i(v) := \begin{cases} 1-c, & \text{if } v_i = 1 \\ 1, & \text{if } v_i = 0 \land \exists j \in N_i, v_j = 1 \\ 0, & \text{if } v_i = 0 \land \forall j \in N_i, v_j = 0 \end{cases} \tag{11}$$

where $0 < c < 1$ is the cost of taking the action (in all experiments $c = \frac{1}{2}$).

For the second set of experiments random games are used, that are not necessarily potential games. These random games were generated with 10 possible strategies for each agent and the utilities in the game matrices of each neighborhood are randomly chosen from a uniform distribution in the range [0,1).

Experimental results

The performance of the best response algorithm and of Algorithm 1 is measured by the social welfare of the final outcome. Social welfare is simply the sum of all utilites.

$$SW(v) = \sum_{i \in N} u_i(v)$$

The first set of experiments uses relatively small games of up to 20 agents in order to compare the performance of both best-response and INEA to the optimal efficiency of stable outcomes which have to be found by exhaustive search. Given our theoretical guarantees on the efficiency of the outcomes of INEA (Corollary 3), the gap from the most efficient outcomes can in principle be unbounded.

The average efficiency of the outcomes of INEA is higher than those obtained by best-response (Figure 2). Moreover, the efficiency of the outcomes of INEA is similar and sometimes better than that of *PNEs of maximal social welfare*. In contrast, best-response converges to outcomes with social welfare that is only slightly better than that of PNEs which minimize the social welfare.

The second set of experiments is performed on large games that are not necessarily potential games and are randomly generated. These games are used to study the dependency of the outcomes of INEA on the transfer function used. An interesting measure is the *fairness* of the outcome (degree of inequality), measured by the *Gini Index*

$$GI(v) = \frac{\sum_{i \in N} \sum_{j \in N} |u_i(v) - u_j(v)|}{2 \sum_{i \in N} \sum_{j \in N} u_j(v)}$$

The fairness of the outcome is highly affected by the type of contracts that are used during search (Figure 3). Outcomes obtained by maximal payoff contracts are less fair than those that use cooperative contracts. When using maximal payoff, multiple agents may secure a payoff transfer to a single change-proposing player which may result in a compensation for the player that is much higher than its benefit from unilateral

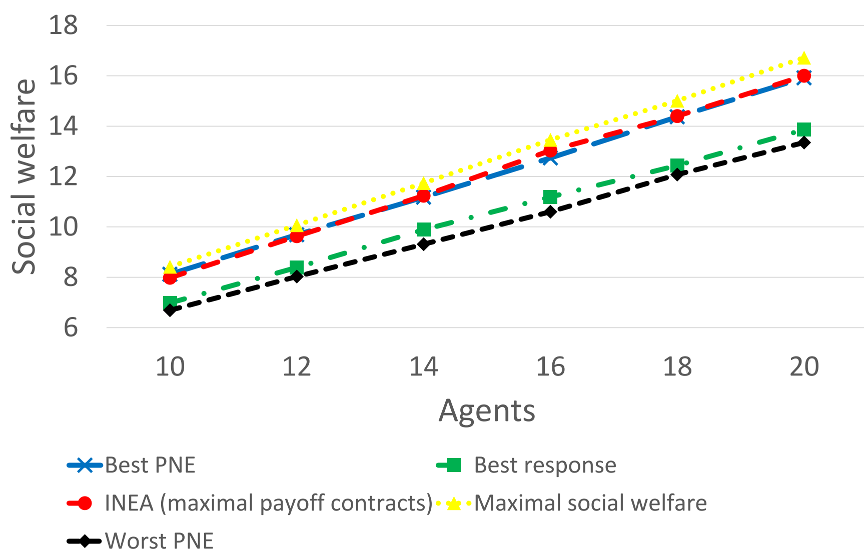

Figure 2: Social welfare of "best-shot" public goods games

deviation. This can lead to a deterioration in the fairness of the solution, as is evident by the increase of the Gini index with the size of the problem, in Figure 3.

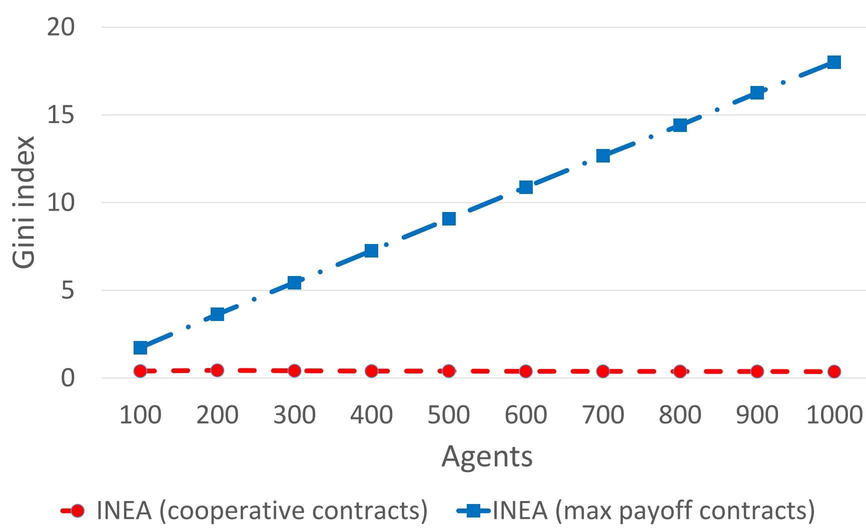

Figure 3: Fairness of stable outcomes on random networks

It is interesting to see that cooperative contracts are also able to provide outcomes of greater efficiency than maximal payoff contracts (Figure 4). When using cooperative contracts the neighbors of a deviating agent may secure less payoff in order to convince it to stay with its current choice. Consequently, a greater improvement in social welfare is possible but needs more improvement steps. The percentage of improvement of social welfare over that of the initial state decreases with problem size. Inspecting Figure 4 one can see that for games with 500 agents the improvement is still sizeable - 20% for maximal payoff contracts and 40% for cooperative contracts.

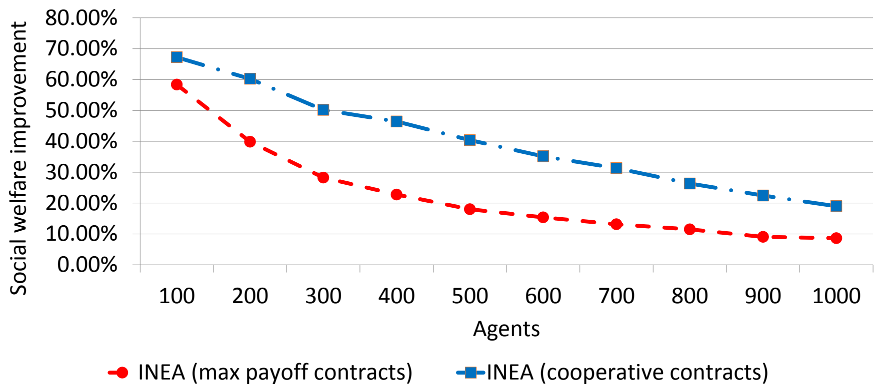

Figure 4: Social welfare of stable outcomes on random networks

Conclusions

An iterative distributed search algorithm for finding a stable outcome of improved efficiency in multi-agents games (MAG) is proposed – the Iterative Nash efficiency Enhancement Algorithm (INEA). The algorithm searches for a solution to the global constraint of stability and performs heuristic distributed optimization search on the soft constraints of efficiency for a good solution (e.g., of higher social welfare).

The INEA algorithm iterates among all agents in the game in a fixed order and each agent in its turn exchanges messages with the agents that interact with it (e.g., its neighbors). Messages propose the selected change of strategy of the agent and in response other agents that are affected by it, may offer transfer of funds in order to compensate the proposer for not changing its strategy and secure a desired outcome.

The INEA algorithm is guaranteed to converge to a state that is of higher efficiency than its initial state and that can be stabilized by the use of transfers of funds among the agents who run the algorithm. A transfer function that can achieve stability is computed by the agents running the algorithm. Since the transfers that secure stability form a binding contract among agents, computed during the search, it is natural to think of them as side payments [13]. The agents playing the game can start from any initial outcome and by running INEA are guaranteed to arrive at a state of higher efficiency that can be stabilized by the use of side payments.

One can think of the proposed INEA algorithm as an extension to the well known best response mechanism (cf. [25, 26]). The proposed method uses transfer of payoffs to extend the standard best-response. The use of this extension guarantees convergence to a stable state in general MAGs, whereas best response is only guaranteed to converge to a PNE for games that are potential games [12]. More importantly, the proposed INEA procedure is guaranteed to converge to a stable state of improved efficiency. This is in contrast to standard best response that does not address efficiency at all.

References

[1] Osborne, M., Rubinstein, A.: A Course in Game Theory. The MIT Press (1994)

[2] DallAsta, L., Pin, P., Ramezanpour, A.: Optimal equilibria of the best shot game. Journal of Public Economic Theory **13**(6) (2011) 885–901

[3] Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.V.: Algorithmic Game Theory. Cambridge University Press (2007)

[4] Nguyen, T.V.A., Lallouet, A.: A complete solver for constraint games. In: Proc. 20th Conf. Princip. Pract. Const. Prog. (CP-2014), Lyon, France (2014) 58–74

[5] Palmieri, A., Lallouet, A.: Constraint games revisited. In: Proc. 26th Intern. Joint Conf. Artif. Intell. (IJCAI-17), Melbourne, AU (2017) 729–35

[6] Roughgarden, T.: Selfish Routing and the Price of Anarchy. The MIT Press (2005)

[7] Monderer, D., Tennenholtz, M.: Strong mediated equilibrium. Artificial Intelligence **173**(1) (2009) 180–195

[8] Grubshtein, A., Meisels, A.: Finding a nash equilibrium by asynchronous backtracking. In: CP. (2012) 925–940

[9] Litov, O., Meisels, A.: Distributed search for pure nash equilibria in graphical games. In: Proc. 2016 Intern. Conf. Auton. Agents & Multiagent Sys. (AAMAS-2016). (2016) 1279–1280

[10] Wahbi, M., Brown, K.N.: A distributed asynchronous solver for nash equilibria in hypergraphical games. In: ECAI. (2016) 1291–1299

[11] Grinshpoun, T., Grubshtein, A., Zivan, R., Netzer, A., Meisels, A.: Asymmetric distributed constraint optimization problems. JAIR **47** (2013) 613–647

[12] Monderer, D., Shapley, L.S.: Potential games. Games and Economic Behavior **14** (1996) 124–143

[13] Jackson, M.O., Wilkie, S.: Endogenous games and mechanisms: Side payments among players. The Review of Economic Studies **72**(2) (2005) 543–566

[14] Kearns, M.J.: Graphical games. In: Vazirani, Vijay V.; Nisan, Noam; Roughgarden, Tim; Tardos, va (2007). Algorithmic Game Theory, Cambridge University Press (2007) 159–178

[15] Jackson, M.O.: Social and economic networks. Volume 3. Princeton University Press Princeton (2008)

[16] Levit, V., Komarovsky, Z., Grinshpoun, T., Meisels, A.: Tradeoffs between incentive mechanisms in Boolean games. In: IJCAI. (2015) 68–74

[17] Jackson, M.O., Zenou, Y.: Games on networks. Handbook of game theory, Peyton Young and Shmuel Zamir, eds. **4** (2014) 102–191

[18] Komarovsky, Z., Levit, V., Grinshpoun, T., Meisels, A.: Efficient equilibria in a public goods game. In: WI-IAT. (2015) 214–219

[19] Grubshtein, A., Meisels, A.: A distributed cooperative approach for optimizing a family of network games. In Brazier, F., K., N., Pavlin, G., Warnier, M., Badica, C., eds.: IDC 2011. Volume 382 of Studies in Computational Intelligence., Delft, the Netherlands, Springer (October 2012) 49–62

[20] Grubshtein, A., Zivan, R., Grinshpoun, T., Meisels, A.: Local search for distributed asymmetric optimization. In: AAMAS. (2010) 1015–1022

[21] Zivan, R., Okamoto, S., Peled, H.: Explorative anytime local search for distributed constraint optimization. Artificial Intelligence **212** (2014) 1–26

[22] Zivan, R., Parash, T., Cohen, L., Peled, H., Okamoto, S.: Balancing exploration and exploitation in incomplete min/max-sum inference for distributed constraint optimization. Autonomous Agents and Multi-Agent Systems **31** (2017) 1165–1207

[23] Shapley, L.: “a value for n-person games”, in h. w. kunh and a. w. tucker, “contributions to the theory of games”. Annals of Mathematical Studies **28** (1953) 307–317

[24] Hirshleifer, J.: From weakest-link to best-shot: The voluntary provision of public goods. Public choice **41**(3) (1983) 371–386

[25] Nisan, N., Schapira, M., Valiant, G., Zohar, A.: Best-response mechanisms. In: Innovations in Computer Science - ICS Beijing, China. (2011) 155–165

[26] Nisan, N., Schapira, M., Valiant, G., Zohar, A.: When is it best to best-respond? SIGecom Exchanges **10** (2011) 16–18