Equilibrium Selection in Potential Games with Noisy Rewards

David Leslie, Jason Marden

Equilibrium Selection in Potential Games with Noisy Rewards

David S. Leslie
School of Mathematics
University of Bristol

Jason R. Marden
Department of Electrical, Computer and Energy Engineering,
University of Colorado at Boulder

***Abstract*—Game theoretical learning in potential games is a highly active research area stemming from the connection between potential games and distributed optimisation. In many settings an optimisation problem can be represented by a potential game where the optimal solution corresponds to the potential function maximizer. Accordingly, significant research attention has focused on the design of distributed learning algorithms that guarantee convergence to the potential maximizer in potential games. However, there are currently no existing algorithms that provide convergence to the potential function maximiser when utility functions are corrupted by noise. In this paper we rectify this issue by demonstrating that a version of payoff-based log-linear learning guarantees that the only stochastically stable states are potential function maximisers even in noisy settings.**

I. Introduction

The current intensive interest in learning in games places great emphasis on designing learning algorithms which will converge to the set of Nash equilibria in classes of games. When the paradigm is used to study distributed optimisation it is usually possible to ensure that the players only need to learn in a potential game [1], [3], [7], [10]. However, the convergence to Nash equilibrium does not actually tell us a great deal about the performance of the optimisation algorithm as a Nash equilibrium could be highly inefficient with regards to the system level objective [11], [12]. In many settings, the potential corresponds to the system level objective so it is desirable to find a learning algorithm which will converge to the global optimum of the potential function [1].

Log-linear learning [2], [8] is a process which is known to converge to the global optimum of the potential function. It does so by the use of simulated-annealing-like transitions between actions [5], although the lack of a central controller means that we lose the ability to insist that exactly one individual updates their action at each time step. However the standard formulation of log-linear learning [2], and many other game-theoretical algorithms, assumes that all players know their own complete reward function (the map from joint action space to the real line) and can observe the actions of all other players in the game. In the very control-theoretic situations in which game theory is supposed to contribute, these assumptions are very unlikely to hold—the games are probably not well-understood in advance, and the distributed nature of the optimisation problem means that observation of all other players is highly problematic.

A payoff-based implementation of log-linear learning has recently been introduced [8] which allows players to learn even when they do not observe opponent actions, and simply respond to received rewards. However this algorithm still requires that the received rewards are deterministic functions of the actions selected by the players which is also unlikely to hold in applications. The received payoff for a particular selection of actions is much more likely to be a random variable with an expected value that is aligned with this deterministic payoff. Hence, the convergence of log-linear learning in this setting is unknown.

Stochasticity in payoffs may arise, not only naturally as a result of stochasticity in the problem, but as a necessary consequence of utility function design [9]. If the utility of a problem is divided among the players using the Shapley value, then calculation of an individual's utility requires a summation over $n!$ terms, where n is the number of players involved in a particular resource. When n is large this is prohibitively expensive, but it is easy to sample utility values with expectation equal to the Shapley value [4]. This sampling to ease computation naturally gives rise to utilities of the form we consider in this article.

In this paper we present an algorithm that extends the payoff-based implementation of log-linear learning [8] to the case of stochastic rewards. The modification of the original formulation is simply to sample repeated observations of the rewards for each joint action instead of observing a single sample; the challenge we address in this article is to show that the inaccurate estimates do not affect the stochastic stability results of the original analysis [8]. Note that a very similar technique could be used to analyse numerous algorithms under a similar modification to accommodate stochastic rewards.

II. Problem statement and algorithm

We consider a finite strategic-form game with player set $\mathcal{I} = \{1, \dots, n\}$. Each player $i \in \mathcal{I}$ has a finite action set $\mathcal{A}_i$ and a utility function $U_i : \mathcal{A} \to \mathbb{R}$ where $\mathcal{A} = \prod_{i \in \mathcal{I}} \mathcal{A}_i$. For an action profile $a = (a_1, a_2, \dots, a_n) \in \mathcal{A}$, let a_{-i} denote the profile of player actions *other than* player i. With this notation, we will sometimes write a joint action $a \in \mathcal{A}$ as (a_i, a_{-i}). Similarly we may write $U_i(a)$ as $U_i(a_i, a_{-i})$. We define player i's *best response set* for an action profile $a_{-i} \in \mathcal{A}_{-i} = \prod_{j \neq i} \mathcal{A}_j$ as

$$B_i(a_{-i}) = \operatorname*{argmax}_{a_i \in \mathcal{A}_i} U_i(a_i, a_{-i}).$$

A Nash equilibrium is any joint action a such that

$$\forall i \in \mathcal{I}, \quad a_i \in B_i(a_{-i}).$$

As indicated in the introduction, many distributed optimisation scenarios may be cast as a potential game. In a potential

game, the change in a player's utility that results from a unilateral change in strategy is equal to the change in the global potential function. Specifically, there exists a function $\phi : \mathcal{A} \to \mathbb{R}$ such that $\forall i \in \mathcal{I}, \forall a_{-i} \in \mathcal{A}_{-i}, \forall a_i, a_i' \in \mathcal{A}_i$,

$$U_i(a_i, a_{-i}) - U_i(a_i', a_{-i}) = \phi(a_i, a_{-i}) - \phi(a_i', a_{-i}).$$

Any joint action a maximising the potential function ϕ is a Nash equilibrium, and so every potential game possesses at least one equilibrium without resorting to mixed strategies.

One way in which distributed optimisation is encoded as a potential game is using the Wonderful Life Utility (WLU) [1], [13], in which individual utility functions are given by

$$U_i(a_i, a_{-i}) = G(a_i, a_{-i}) - G(\mathring{a}_i, a_{-i})$$

where $\mathring{a}_i$ is an arbitrary reference action of player i and G is the global system utility. If the reference action corresponds to 'do nothing' then the WLU payoff to player i corresponds to the marginal contribution to the global utility made by i. For this form of utility it is easily verified that the global utility G acts as a potential function for the game. It is therefore important to know whether the players can reach the global optimum of the potential function of such a game.

Log-linear learning (LLL) [2], [8] is one of very few algorithms that achieves selection of the equilibrium corresponding to the potential function maximiser. In particular, LLL guarantees that only the joint actions that maximise the potential function are stochastically stable. However all current implementations of LLL assume either that the utility functions are known in advance, or (for payoff-based LLL [8] as described in Algorithm 1) that players observe uncorrupted values of this utility function. In real life implementations of game-theoretical learning for optimisation it is likely that the utility functions are not known in advance, and that observed utilities are subjected to noise. We encapsulate this feature of real distributed optimisation in the following modelling assumption.

Assumption (Noisy rewards): When players select joint action $a \in \mathcal{A}$, each player i receives a reward $R_i = U_i(a) + \xi_i$ where the perturbations ξ_i have expectation 0, variance $V_i(a)$, and are independent of all other random variables.

We therefore require a variant of log-linear learning that can accommodate the stochasticity in the received rewards. Indeed, the algorithm we propose (Algorithm 2) is a direct modification of payoff-based LLL [8] in which the players sample multiple copies of the rewards for each action instead of using a single observation. In our modification, a time counter t no longer indexes single plays of the game, but instead indexes blocks of plays of length $2N$. In the first half of each block the players repeat action $a(t-1)$ for N further plays to get a new estimate $\hat{U}_i^t(a(t-1))$ of $U_i(a(t-1))$; in the second half of block the players play action $a(t)$ for N iterations to obtain an estimate $\hat{U}_i^t(a(t))$ of $U_i(a(t))$. Let $x_i(t)$ be the indicator of whether Player i experiments on block t. If $x_i(t-1) = 0$ then either (with high probability) $a_i(t) = a_i(t-1)$ and Player i doesn't experiment ($x_i(t) = 0$); otherwise $a_i(t)$ is selected uniformly at random from $|\mathcal{A}_i|$ (Player i experiments: $x_i(t) = 1$). If $x_i(t-1) = 1$, so that $a_i(t-1)$ was experimental, then action $a_i(t)$ is selected according to a logistic function of the estimates $\hat{U}_i^{t-1} = (\hat{U}_i^{t-1}(a(t-2)), \hat{U}_i^{t-1}(a(t-1)))$ and $x_i(t) = 0$. Note that the advantage of such a *payoff-based* scheme is that each player does not need to have information regarding the other players of the game or even their own payoff function—each player simply updates their current action according to the rewards they observe.

Algorithm 1 Payoff based log-linear learning [8]

Fix parameters τ, ω, and for each $i \in \mathcal{I}$ set $x_i(0) = 0$ and select $a_i(0)$ arbitrarily from $\mathcal{A}_i$.
For each $t \in \mathbb{N}$, each $i \in \mathcal{I}$ carries out the following:
if $x_i(t-1) = 0$ **then** {player i did not experiment}
With probability $1-\omega$ set $x_i(t) = 0$, $a_i(t) = a_i(t-1)$
Otherwise set $x_i(t) = 1$ and select $a_i(t)$ uniformly at random from $\mathcal{A}_i$.
else {i.e. if $x_i(t-1) = 1$, player i did experiment}
Set $x_i(t) = 0$, and set

$$a_i(t) = \begin{cases} a_i(t-2) & \text{w.p. } \frac{e^{\frac{1}{\tau}U_i(a(t-2))}}{e^{\frac{1}{\tau}U_i(a(t-2))}+e^{\frac{1}{\tau}U_i(a(t-1))}} \\ a_i(t-1) & \text{w.p. } \frac{e^{\frac{1}{\tau}U_i(a(t-1))}}{e^{\frac{1}{\tau}U_i(a(t-2))}+e^{\frac{1}{\tau}U_i(a(t-1))}} \end{cases}$$

end if

Algorithm 2 Sampled payoff based log-linear learning

Fix parameters τ, ω, N, and for each $i \in \mathcal{I}$ set $x_i(0) = 0$ and select $a_i(0)$ arbitrarily from $\mathcal{A}_i$.
For each $t \in \mathbb{N}$, each $i \in \mathcal{I}$ carries out the following:
if $x_i(t-1) = 0$ **then** {player i did not experiment}
With probability $1-\omega$ set $x_i(t) = 0$, $a_i(t) = a_i(t-1)$
Otherwise set $x_i(t) = 1$ and select $a_i(t)$ uniformly at random from $\mathcal{A}_i$.
else {i.e. if $x_i(t-1) = 1$, player i did experiment}
Set $x_i(t) = 0$, and set

$$a_i(t) = \begin{cases} a_i(t-2) & \text{w.p. } \frac{e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-2))}}{e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-2))}+e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-1))}} \\ a_i(t-1) & \text{w.p. } \frac{e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-1))}}{e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-2))}+e^{\frac{1}{\tau}\hat{U}_i^{t-1}(a(t-1))}} \end{cases}$$

end if
Play action $a_i(t-1)$ for N plays of the game, and let $\hat{U}_i^t(a(t-1))$ be the average reward obtained.
Play action $a_i(t)$ for N plays of the game, and let $\hat{U}_i^t(a(t))$ be the average reward obtained.

Note that, even with $N = 1$, this algorithm differs slightly from Algorithm 1, in that within 'block' t we start by repeating action $a(t-1)$ before playing action $a(t)$. The reason for this repetition is so that we can write down a transition probability on a discrete state space (comprising of actions and experimental status x only). If the action value estimates from block $t-1$ were to be carried forward from block $t-1$ into block $t+1$ to be used for action selection then the state

space of the Markov chain becomes more complicated. We have no doubt that very similar conclusions could be drawn if this repetition were not to be carried out, but for the sake of brevity, and to retain the framework of discrete state space stochastic stability, we modify the algorithm to ensure that we have a discrete space Markov chain.

III. Convergence of Algorithm 1 without stochastic rewards

In this section we prove convergence of Algorithm 2 when there is no stochasticity in the received rewards. Under this scenario $\hat{U}_i^t(a(t-1)) \equiv U_i(a(t-1))$ and $\hat{U}_i^t(a(t)) \equiv U_i(a(t))$.

Proposition 1: Consider any finite n-player potential game with potential function $\phi : \mathcal{A} \to \mathbb{R}$ where all players adhere to Algorithm 2. Set $\omega = (\mathrm{e}^{-1/\tau})^m$. If $V_i(a) = 0$ for all i and a (i.e. rewards are deterministic) then for sufficiently large m the stochastically stable states are contained in the set of potential function maximisers.

Proof: Following [8], write $z(t) = [a(t-1), a(t), x(t)] \in \mathcal{A} \times \mathcal{A} \times \{0,1\}^n$ for the state of the Markov chain, $\epsilon = \mathrm{e}^{-1/\tau}$, and $P^{\epsilon}_{z(t)\to z(t+1)}$ for the transition probabilities of the chain. Suppose that $z(t) = [a(t-1), a(t), x(t)] \to z(t+1) = [a(t), a(t+1), x(t+1)]$ is a valid transition of the chain (i.e. $x_i(t) = x_i(t+1) = 0 \Rightarrow a_i(t+1) = a_i(t)$, $x_i(t) = 1 \Rightarrow (x_i(t+1) = 0$ and $a_i(t+1) \in \{a_i(t-1), a_i(t)\})$). As in the proof of Claim 6.1 of [8], the transition probability is given by

$$
\begin{aligned}
&P^{\epsilon}_{z(t)\to z(t+1)} = \\
&\left[\prod_{i:x_i(t)=0,x_i(t+1)=0} (1-\omega)\right]\left[\prod_{i:x_i(t)=0,x_i(t+1)=1} \frac{\omega}{|\mathcal{A}_i|}\right] \\
&\times\left[\prod_{i:x_i(t)=1,a_i(t+1)=a_i(t-1)} \frac{\epsilon^{-U_i(a(t-1))}}{\epsilon^{-U_i(a(t-1))}+\epsilon^{-U_i(a(t))}}\right] \\
&\times\left[\prod_{i:x_i(t)=1,a_i(t+1)=a_i(t)} \frac{\epsilon^{-U_i(a(t))}}{\epsilon^{-U_i(a(t-1))}+\epsilon^{-U_i(a(t))}}\right].
\end{aligned} \tag{1}
$$

Since this is identical to the transition matrix of the original payoff-based implementation the conclusion of Theorem 6.1 of [8] continues to hold. ∎

IV. Sampled LLL has the same limit behaviour

We now re-introduce stochasticity in the received rewards. To ease notation, throughout this section we fix t, and recall that $\hat{U}_i^t = (\hat{U}_i^t(a(t-1)), \hat{U}_i^t(a(t))$. Note that the only resulting change in the transition probabilities of the Markov chain when compared with the deterministic case is in the decision of player i over which action to select at time $t+1$ if $x_i(t) = 1$ (i.e. the player experimented on the previous block). We will show that by increasing the number of samples N at an appropriate rate as $\tau \to 0$ the stochastically stable states of the learning process are not changed.

Proposition 2: Consider any finite n-player potential game with potential function $\phi : \mathcal{A} \to \mathbb{R}$ where all players observe noisy rewards (as in the assumption above) and adhere to Algorithm 2. Set $\omega = (\mathrm{e}^{-1/\tau})^m$ and set

$$
N = \frac{g(\tau)(1 + e^{\Delta_{\max}/\tau})}{f(\tau)^2\tau^2}, \tag{2}
$$

where f is any function that tends to 0 as $\tau \to 0$, g is any function that tends to ∞ as $\tau \to 0$, and $\Delta_{\max}$ is the maximum difference in reward between any two actions for any player. Then for sufficiently large m the stochastically stable states are contained in the set of potential function maximisers.

Proof: The method we will use is to show that transitions of Algorithm 2 have the same resistance [14] as the transitions when $V_i(a) \equiv 0$ (i.e. the situation covered by Prop. 1). Consider equation (1); in particular notice that the terms

$$
k_i(a(t); z(t)) := \frac{\epsilon^{-U_i(a(t))}}{\epsilon^{-U_i(a(t-1))} + \epsilon^{-U_i(a(t))}}
$$

are simply $\mathbb{P}(a_i(t+1) = a_i(t) \,|\, z(t))$ where the probability is calculated for when there is no stochasticity. (In what follows, whatever is said for $k_i(a(t); z(t))$ also holds for the equivalently defined $k_i(a(t-1), z(t))$ but we do not repeat the analysis.) In the version where we use estimates $\hat{U}_i^t$ we need to take an expectation over these sampled values to calculate the corresponding probability, and thus the k_i terms in (1) are replaced by terms

$$
\tilde{k}_i(a(t); z(t)) := \mathbb{E}\left[\frac{\epsilon^{-\hat{U}^t(a(t))}}{\epsilon^{-\hat{U}^t(a(t-1))} + \epsilon^{-\hat{U}^t(a(t))}} \,\middle|\, z(t)\right].
$$

We will show that

$$
\frac{\tilde{k}_i(a(t); z(t))}{k_i(a(t); z(t))} \to 1 \quad \text{as} \quad \tau \to 0. \tag{3}
$$

Define $\tilde{P}^{\tau}_{z^0\to z^1}$ to be the transition probability from state z^0 to state z^1 when sampling is used. Under (3), $\forall \eta > 0$

$$
\tilde{P}^{\tau}_{z^0\to z^1} = \frac{\tilde{P}^{\tau}_{z^0\to z^1}}{P^{\tau}_{z^0\to z^1}} P^{\tau}_{z^0\to z^1} \begin{cases} \leq & (1+\eta)P^{\tau}_{z^0\to z^1} \\ \geq & (1-\eta)P^{\tau}_{z^0\to z^1} \end{cases}
$$

for sufficiently small τ (since the ratio tends to 1). Hence $\tilde{P}^{\tau}_{z^0\to z^1} \to P^{\tau}_{z^0\to z^1}$, the resistance [14] of each transition is the same under Algorithm 2 as under Algorithm 1, and the conclusion of Prop. 1 continues to hold. To ease notation, for the rest of the proof we condition all probability statements on $z(t)$

To show (3), start by noting that the action selection decision (and in particular k or $\tilde{k}$) depends only on the difference between rewards, and we define

$$
\begin{aligned}
\Delta_i &:= U_i(a(t)) - U_i(a(t-1)), \\
\hat{\Delta}_i &:= \hat{U}_i^t(a(t)) - \hat{U}_i^t(a(t-1))
\end{aligned}
$$

so that

$$
k_i(a(t); z(t)) = (1 + \epsilon^{\Delta_i})^{-1}. \tag{4}
$$

By our assumption that the random rewards received by player i on each play of the game are independent, with expectation $U_i(a)$ and variance $V_i(a)$ when joint action a is played,

$$
\mathbb{E}(\hat{\Delta}_i) = \Delta_i \text{ and } \mathrm{Var}(\hat{\Delta}_i) = \frac{V}{N} \tag{5}
$$

where $V = V_i(a(t)) + V_i(a(t-1))$.

Now consider the event

$$A^\delta := \{|\Delta_i - \hat{\Delta}_i| < \delta\}, \tag{6}$$

under which the estimated difference between the actions is close to the true difference. Clearly, when A^δ occurs, the transition probabilities under sampling are close to those under perfect reward information. We will show that we can increase the probability of A^δ sufficiently quickly to ensure that $\tilde{k}$ overall is close to k.

First note that by Chebychev's inequality (see for example [6])

$$\mathbb{P}(\bar{A^\delta}) \leq \frac{V}{N\delta^2} \tag{7}$$

where $\bar{A^\delta}$ is the complement of A^δ. We now use the partition theorem of probability to see that

$$\begin{aligned}\tilde{k}_i(a(t);z(t)) &= \mathbb{P}(a_i(t+1) = a_i(t) \,|\, A^\delta)\mathbb{P}(A^\delta) \\ &\quad + \mathbb{P}(a_i(t+1) = a_i(t) \,|\, \bar{A^\delta})\mathbb{P}(\bar{A^\delta}).\end{aligned}$$

Hence

$$\begin{aligned}&|\tilde{k}_i(a(t);z(t)) - k_i(a(t);z(t))| \\ &\leq \left|\mathbb{P}(a_i(t+1) = a_i(t) \,|\, A^\delta) - k_i(a(t);z(t))\right| \mathbb{P}(A^\delta) \\ &\quad + \left|\mathbb{P}(a_i(t+1) = a_i(t) \,|\, \bar{A^\delta}) - k_i(a(t);z(t))\right| \mathbb{P}(\bar{A^\delta})\end{aligned}$$

The final line is bounded above by $2V/(N\delta^2)$, by (7) and noting that probabilities are less than 1. Note also that the absolute value of the derivative of (4) with respect to Δ_i is $\tau^{-1}k_i(a(t);z(t))(1-k_i(a(t);z(t))) \leq \tau^{-1}k_i(a(t);z(t))$ and under A^δ we have $|\Delta_i - \hat{\Delta}_i| < \delta$, so by the mean value theorem

$$\left|\mathbb{P}(a_i(t+1) = a_i(t) \,|\, A^\delta) - k_i(a(t);z(t))\right| \leq \frac{\delta}{\tau} k_i(a(t);z(t)).$$

Therefore

$$\frac{\left|\tilde{k}_i(a(t);z(t)) - k_i(a(t);z(t))\right|}{k_i(a(t);z(t))} \leq \frac{\delta}{\tau} + 2\frac{V}{N\delta^2 k(a(t);z(t))}$$

and to prove (3) it suffices to show that the right hand side of this expression tends to 0. In particular if we can show that

$$\frac{\delta}{\tau} \to 0, \quad \text{and} \tag{8}$$

$$N\delta^2 k_i(a(t);z(t)) \to \infty \tag{9}$$

then the result is proved.

Note that δ is chosen for our convenience in this proof, so let $\delta = \tau f(\tau)$, where f is any function that tends to 0 as $\tau \to 0$, and (8) will be satisfied. Now note that $k(a(t);z(t)) \geq (1+\epsilon^{\Delta_{\max}})^{-1}$ where $\Delta_{\max}$ is the maximal utility difference between two actions; therefore (9) holds if

$$N = \frac{g(\tau)(1+e^{\Delta_{\max}/\tau})}{f(\tau)^2\tau^2},$$

where g is any function that tends to infinity as $\tau \to 0$. Hence by suitable choice of N we can ensure that the resistance of any transition under Algorithm 2 is the same as the transition when there is no stochasticity. The result of Prop. 1 therefore continues to hold. ■

V. Conclusion

We have extended the method of log-linear learning to the situation where players only observe noisy evaluations of their payoffs and cannot observe actions selected by 'opponents'. We have modified the original proof that only potential function maximisers are stochastically stable [8] by showing that transitions under sampled reward values have the same resistance as transitions of the algorithm when there is no stochasticity, provided that the number of samples increases at a suitable rate. This is therefore the first algorithm that is proved to select the Nash equilibrium corresponding to the maximum of the potential function under noisy reward observations.

This result is important for the application of learning in games to distributed optimisation problems, since distributed optimisation resorts to game theory in precisely the situations where the 'game' is not well-defined in advance, and when payoffs are subject to stochastic shocks. Since standard results on learning in games usually assume, as a minimum, that players receive a payoff equal to the deterministic utility function of the played joint action, the extension to stochastic rewards is an important bridge between the theoretical convergence results and the application scenarios of interest.

References

[1] Arslan, G., J. R. Marden, and J. S. Shamma (2007). Autonomous vehicle-target assignment: a game theoretical formulation. *Journal of Dynamic Systems, Measurement, and Control*, *129*, 584–596.

[2] Blume, L. E. (1993). The statistical mechanics of strategic interaction. *Games and Economic Behavior*, *5*, 387–424.

[3] Chapman, A. C., A. Rogers, N. R. Jennings, and D. S. Leslie (forthcoming). A unifying framework for iterative approximate best response algorithms for distributed constraint optimisation problems. *Knowledge Engineering Review.*

[4] Conitzer, V. and T. Sandholm (2004). Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. *AAAI 2004*, AAAI press.

[5] Geman, S. and D. Geman (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machince Intelligence*, *5*, 721–741.

[6] Grimmett, G. R. and D. R. Stirzaker (1982). *Probability and Random Processes*. Oxford University Press.

[7] Marden, J. R., G. Arslan, and J. S. Shamma (2009). Cooperative control and potential games. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *39*, 1393–1407.

[8] Marden, J. R. and J. S. Shamma (2008). Revisiting log-linear learning: Asynchrony, completeness and payoff-based implementation. *Submitted to Games and Economic Behavior*.

[9] Marden, J. R. and A. Wierman (2008). Distributed welfare games. *Proceedings of the 47th IEEE Conference on Decision and Control*.

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

[11] Nisan, N., T. Roughgarden, E. Tardos, and V. V. Vazirani (eds) (2007). *Algorithmic Game Theory*. Cambridge University Press.

[12] Roughgarden, T. (2005). *Selfish Routing and the Price of Anarchy*. MIT Press.

[13] Wolpert, D. and K. Tumor (1999). An overview of collective intelligence. In J. M. Bradshaw, editor, *Handbook of Agent Technology*. AAAI press/MIT press.

[14] Young, H. P. (1993). The evolution of conventions. *Econometrica 61*, 57–84.

[15] Young, H. P. (2004). *Strategic Learning and its Limits*. Oxford University Press.