Finding Bounded Rational Equilibria Part I: Iterative Focusing

David H. Wolpert*

February 4, 2005

Abstract

A long-running difficulty with conventional game theory has been how to modify it to accommodate the bounded rationality characterizing all real-world players. A recurring issue in statistical physics is how best to approximate joint probability distributions with decoupled (and therefore far more tractable) distributions. It has recently been shown that the same information theoretic mathematical structure, known as Probability Collectives (PC) underlies both issues. This relationship between statistical physics and game theory allows techniques and insights from the one field to be applied to the other. In particular, PC provides a formal model-independent definition of the degree of rationality of a player and of bounded rationality equilibria. This pair of papers extends previous work on PC by introducing new computational approaches to effectively find bounded rationality equilibria of common-interest (team) games.

1 INTRODUCTION

The fields of statistical physics, game theory, and distributed control/optimization share one fundamental characteristic: they are all concerned with how the probability distribution governing a distributed system is related to the functionals that it optimizes. This shared characteristic provides the basis for a mathematical language for translating many of the concepts of those fields into one another. This mathematical language is known as Probability Collectives (PC) [1, 2, 3, 4, 5, 6]. By allowing us to transfer theory and techniques between those fields, it provides a means of unifying them.

This pair of papers introduces computational techniques from PC for efficiently finding bounded rational equilibria of noncooperative games. In this paper first we review PC and how to use it to formalize bounded rationality. We then review two of the previously explored techniques for finding bounded rational equilibria, Brouwer updating and Nearest Newton updating. After this we

*D. Wolpert is with NASA Ames Research Center, Moffett Field, CA, 94035 dhw@ptolemy.arc.nasa.gov

introduce iterative focusing, a new set of techniques for finding full rationality equilibria.

Due to space limitations, several other schemes for finding bounded rational equilibria could not be presented in this first paper. They are instead introduced in the second paper [7]. That second paper also shows how to extend all of the approaches for finding equilibria (from both papers) to the case of uncountable move spaces of the players. Some issues that arise in practice when running these algorithms are also discussed there.

The version of Probability Collectives considered in this paper, involving product distributions, is called "Product Distribution" (PD) theory[1]. It's important to note that PD theory also has many applications in science beyond those considered in this paper. For example, see [3, 4, 8, 9, 10, 5, 6, 11] for work concerning distributed control and to distributed optimization. See also [12, 13, 10] for work showing, respectively, how to use PD theory to improve Metropolis-Hastings sampling, how to relate it to the mechanism design work in [14, 15, 16, 17], and how to extend it to continuous move spaces and time-extended strategies.

Throughout these papers δ functions are either Dirac or Kronecker as appropriate, integrals implicitly have a measure appropriate to the cardinality of the underlying space, and Θ is the Heaviside step function.

2 Review of Probability Collectives

2.1 Game theory and Statistical Physics

In noncooperative game theory one has a set of N players, each choosing its (normal form) strategy x_i independently, by sampling a distribution $q_i(x_i)$ over those strategies. Each player i also has her own cost function $g_i(x)$, specifying how much punishment she gets for every possible joint-strategy x of all N players.[1] Let $q_{-i}(x_{-i})$ mean the joint probability distribution of all players other than i, i.e., $\prod_{j \neq i} q_j(x_j)$. Then the "goal" of each player i is to set q_i so that, when conditioned on q_{-i}, the expected value of i's cost is as low as possible.

Conventional noncooperative ngame theory assumes each player i is "fully rational", i.e., able to solve for that optimal q_i, and that she uses that optimal distribution. It is primarily concerned with analyzing such equilibria of the game [18, 19, 20, 21]. In the real world, this assumption of full rationality almost never holds, whether the players are humans, animals, or computational agents [22, 23, 24, 25, 26, 27, 28, 29, 30]. This is due to the cost of computation of that optimal distribution, if nothing else. This real-world bounded rationality is one of the major impediments to applying conventional game theory in the real world.

[1] A "cost" function is just the negative of a utility function. We work with costs rather than utilities to agree with usage in the optimization and physics communities, where the goal is minimization of objective functions rather than maximization.

Perhaps the most succinct and principled way of deriving statistical physics is as an application of the Maximum Entropy (Maxent) principle of information theory [31, 32, 33]. In this formulation, the problem of statistical physics is cast as how best to infer the probability distribution over a system's states when one's prior knowledge consists purely of the expectation values of certain functions of the system's state [34, 33]. For example, this prescription says we should infer that the probability distribution p governing the system is the Boltzmann distribution when our prior knowledge is the system's expected energy. This scenario known as the "canonical ensemble"; other ensembles arise when other expectation values are added to one's prior knowledge. In particular, if the number of particles of various types is uncertain, but one knows their expectation values, one arrives at the "grand canonical ensemble".

It has recently been recognized that this exact approach used in statistical physics provides a principled way to modify conventional game theory to accommodate bounded rationality [2]. Doing so results in the principle that the bounded rational equilibrium is the minimizer of a certain set of coupled Lagrangian functions of the joint distribution, $q(x) = \prod_i q_i(x_i)$ (see below). The resultant distributions are the same ones that arise in the canonical ensemble of statistical physics. In particular, the Nash equilibrium of a game simply corresponds to the special case in statistical physics where the temperature equals 0. Bounded rationality corresponds to non-zero temperature.

In addition to showing how to formulate bounded rationality, PC provides many other benefits to game theory. For example, its formulation of bounded rationality explicitly includes a term that, in light of information theory, is naturally interpreted as a cost of computation. So from the point of view of PC, it is formally correct to say that bounded rationality arises due to the cost of computing the Nash equilibrium. As another example, one can apply the grand canonical ensemble to game theory rather than the canonical ensemble. This extends game theory to allow for stochastic numbers of players of various types [1] in addition to bounded rationality of those players. Among other things, this results in some novel extensions of evolutionary game theory.[2] As a final example, in [2] some algorithms for finding bounded rational equilibria are derived that are related to Stackelberg games. More generally, those algorithms are related to the problem of finding the optimal control hierarchy for team of players with a common goal, i.e., finding an optimal organization chart.

2.2 The Maxent Lagrangian

In a nutshell, the perspective adopted by the PC approach to game theory is purely empiricist — you are a scientist, with limited information about a particular game, and must predict what strategies are being followed in the game. The prediction you make is the associated equilibrium concept.

[2] It's intersting to note that the replicator dynamics of evolutionary game theory also arises in PC for games with a fixed number of players. It turns out that it gives the infinitesimal time limit of certain 2nd order descent schemes for finding bounded rational equilibria of such games [13].

Making such a prediction is an exercise in statistical inference, an old and well-understood field. No modeling of the players and their thought processes need arise in such inference. Of course, if we are provided hard data (!) concerning how the players behave, those should be taken into account in the inference. In addition, models concerning the behaviors of players can be used, if desired, for example via Bayesian inference. Note that any priors arising in that inference are set by *us*, the scientists predicting the game, not by the players involved in the game. In contrast, if we believe the players to be Bayesians, that simply means that our inference algorithm must involve probability distributions over the possible priors of the players.

To formalize this, for simplicity assume a a finite number of possible moves (pure strategies) $|X_i|$ for each player i. Also for simplicity, consider a common payoff (aka team) game, in which all players share the same cost function, $G(x)$. (The more general case is addressed in [2].) Say that we are told the expected value of G. So our prior knowledge is that the players are independent, and that their expected cost is some ϵ. The Maxent principle of information theory tells us that the optimal estimate of the q for that prior knowledge is given by the minimizer of the **Maxent Lagrangian**

$$\begin{aligned} \mathscr{L}(q) &\triangleq \beta[E_q(G) - \epsilon] - S(q) \\ &= \beta[\int dx \, \prod_j q_j(x_j) G(x) - \epsilon] - S(q) \end{aligned} \tag{1}$$

where β is the Lagrange parameter implicitly set by the constraint on the expected cost [1, 2, 9] [3].

Solving, we find that the mixed strategies minimizing the Maxent Lagrangian are related to each other via the coupled **Boltzmann distributions**

$$q_i(x_i) \propto e^{-E_{q_{-i}}(G|x_i)}. \tag{2}$$

Following Nash, we can use Brouwer's fixed point theorem to establish that for any fixed value β, there must exist at least one product distribution q solving these equations.

The first term in $\mathscr{L}$ is minimized by perfectly rational players. The second term is minimized by perfectly *irrational* players, i.e., by uniform mixed strategies q_i. So β in the maxent Lagrangian explicitly specifies the balance between the rational and irrational behavior of the player. In particular, for $\beta \to \infty$, by minimizing the Lagrangian we recover the Nash equilibria of the game: in that limit the set of q that minimize the Lagrangian is the same as the set of delta functions about the Nash equilibria of the game.

See [2] for discussion of how the Maxent Lagrangian encompasses cost of computation and its extension to non-common-payoff games. Also discussed there is the extension to other kinds of prior knowledge, to other kinds of statistical inference (e.g., Bayesian inference), and to variable numbers of players.

[3] Throughout this paper the terms in any Lagrangian that restrict distributions to sum to one are implicit. The other constraint, that none of its components are negative, will not need to be explicitly enforced.

Also in that work is a discussion relating the Maxent Lagrangian and its derivation to other work in the game theory community. See [35] for a discussion of the relationship with other techniques in the optimization community.

2.3 Descent of the Maxent Lagrangian

Say we wish to find a bounded rational equilibrium, i.e., a minimizer of the Maxent Lagrangian, and that to do this we are iteratively evolving q to minimize $\mathscr{L}$ for some fixed β. Say at the current stage of this process we are at some point $q \in \mathcal{Q}$. It is proven in [9] that the direction from q within $\mathcal{Q}$ that, to first order, results in the largest drop in the value of $\mathscr{L}(q)$, is as follows:

$$\frac{\partial^R \mathscr{L}(q)}{\partial^R q_i(x_i = j)} \quad = \quad u_i(j) - \sum_{x'_i} u_i(x'_i)/|X_i|, \tag{3}$$

where $u_i(j) \triangleq \beta E(G \mid x_i = j) + \ln[q_i(j)]$, and the symbol ∂^R indicates that we do not mean the indicated partial derivative, formally speaking, but rather the indicated component of the 1st-order descent vector [4].

Eq. 3 gives the (negative of the) change that each agent should make to its distribution to have them jointly implement a step in steepest descent of the Maxent Lagrangian. These updates are completely distributed, in the sense that each agent's update at time t is independent of any other agents' update at that time. Typically at any t each agent i knows $q_i(t)$ exactly, and therefore knows $\ln[q_i(j)]$. However often it will not know G and/or the q_{-i}. In such cases it will not be able to evaluate the $E(G \mid x_i = j)$ terms in Eq. 3 in closed form.

One way to circumvent this problem is to have those expectation values be simultaneously estimated by all agents by repeated Monte Carlo sampling of q to produce a set of $(x, G(x))$ pairs. Those pairs can then be used by each agent i to estimate the values $E(G \mid x_i = j)$, and therefore how it should update its distribution. In the simplest version of this scheme such an update to q only occurs once every L time-steps. Note that only one set of Monte Carlo samples is needed for all players to determine how to update their mixed strategy, no matter how many players there are.

In this simple Monte Carlo scheme only the samples $(x, G(x))$ formed within a block of L successive time-steps are used at the end of that block by the agents to update their distributions (according to Eq. 3). More sophisticated approaches re-use old samples, modify the G values returned by the Monte Carlo on a player-by-player basis, etc. [36, 1, 7, 10]. In addition, in general it may be for agent i that it has no sample for the value x_i. These scenarios are addressed in [7], the second paper in this series.

[4] Formally speaking, the partial derivative is given by $u_i(j)$. Intuitively, the reason for subtracting $\sum_{x'_i} u_i(x'_i)/|X_i|$ is to keep the distribution in the set of all possible probability distributions over x, $\mathcal{P}$.

2.4 Brouwer updating

There are many other descent schemes in addition to ones related to steepest descent. In Brouwer updating for the team game, an agent i adopts the distribution q_i given by Eq. 2 using the current q_{-i}. This is the update that would minimize the Maxent Lagrangian *if all other agents did not change their distributions.* Since this update rule only depends on $E(G \mid x_i)$, it can be estimated using Monte Carlo samples in the usual way.

In **serial** Brouwer updating, only one agent at a time performs the update. The order in which the agents update their distributions can be pre-fixed or random. It can also be dynamically determined in a greedy manner, by choosing which agent to update based on what associated drop in the Maxent Lagrangian would ensue [1, 13, 5]. (These various ordering schemes are similar to the those used in the majorization and block relaxation techniques of in optimization statistics.) Aside from Monte Carlo estimation error, each step of serial Brouwer updating is guaranteed not to increase the associated (Maxent) Lagrangian. Note that with serial Brouwer updating, the agents are in effect playing a sequence of Stackelberg games[19, 18].

In **parallel** Brouwer updating, this procedure is followed simultaneously by all agents. Accordingly, parallel Brouwer updating can be viewed as a variant of ficticious play (see [37, 38, 39]). In mixed serial-Brouwer updating, in a given iteration one performs a sequence of more than one Brouwer update, each update being applied in parallel to a subset of all the agents. Such updating can be viewed as a management hierarchy in a human organization, in which at each iteration a subset of the people in the organization make their decisions in parallel with one another, and are then followed in their decision-making by other such subsets of people.

Now in general, when any q_j changes, for every $i \neq j$, what distribution q_i minimizes i's Lagrangian will change, in accord with Eq. 2. This suggests that a step of parallel Brouwer updating may "thrash", and have each agent change in a way that confounds the other agents' changes. (See [37, 39] and references therein for analysis of related issues in ficticious play.) In such a case the update may not actually decrease the associated (Maxent) Lagrangian, unlike with serial Brouwer.

There are many possible ways of mixing parallel and serial Brouwer updating, in which only subsets of the agents perform parallel updates at any given time. These can be viewed as management hierarchies, akin to those in human organizations. Often such hierarchies can also be determined dynamically, during the updating process.

2.5 Kullback-Leibler distance

Say that we were not restricting ourselves to product distributions. So the Lagrangian becomes $\mathscr{L}(p) = \beta(E_p(G) - \gamma) - S(p)$, where p can now be any distribution over x. There is only one local minimum over p of this Lagrangian, the **canonical ensemble**:

$$p^\beta(x) \propto e^{-\beta G(x)}$$

In general p^β is not a product distribution. However we can ask what product distribution is closest to it. Now in general, the proper way to approximate a **target distribution** p with a distribution from a subset $\mathcal{C}$ of the set of all distributions is to first specify a misfit measure saying how well each member of $\mathcal{C}$ approximates p, and then solve for the member with the smallest misfit. This is just as true when $\mathcal{C}$ is the set of all product distributions as when it is any other set.

How best to measure distances between probability distributions is a topic of ongoing controversy and research [40]. The most common way to do so is with the infinite limit log likelihood of data being generated by one distribution but misattributed to have come from the other. This is know as the **Kullback-Leibler distance** [31, 41, 32]:

$$KL(p_1 \mid\mid p_2) \triangleq S(p_1 \mid\mid p_2) - S(p_1) \tag{4}$$

where $S(p_1 \mid\mid p_2) \triangleq -\int dx \; p_1(x)\ln[\frac{p_2(x)}{\mu(x)}]$ is known as the **cross entropy** from p_1 to p_2 (and as usual we implicitly choose uniform μ). The KL distance is always non-negative, and equals zero iff its two arguments are identical.

As shorthand, define the "pq distance" as $KL(p \mid\mid q)$, and the "qp distance" as $KL(q \mid\mid p)$, where p is our target distribution and q is a product distribution. Then it is straightforward to show that the qp distance from q to target distribution p^β is just the Maxent Lagrangian $\mathscr{L}(q)$, up to irrelevant overall additive constants. In other words, the q minimizing the Maxent Lagrangian is the same as the q having minimal qp distance to the associated canonical ensemble.

However the qp distance is the (infinite limit of the negative log of) the likelihood that distribution p would attribute to data generated by distribution q. It can be argued that a better measure of how well q approximates p would be based on the likelihood that q attributes to data generated by p. This is the pq distance; it gives a different Lagrangian from that of Eq. 1.

Evaluating, up to an overall additive constant (of the canonical distribution's entropy), the pq distance is

$$KL(p^\beta \mid\mid q) = -\sum_i \int dx \; p(x)\ln[q_i(x_i)].$$

This is equivalent to a game where each coordinate i has the "Lagrangian"

$$\mathscr{L}^*_i(q) \quad \triangleq \quad -\int dx_i \; p^\beta_i(x_i)\ln[q_i(x_i)], \tag{5}$$

where $p^\beta_i(x_i)$ is the marginal distribution $\int dx_{-i} p^\beta(x)$. Since q is a product distribution, the minimizer of this is just $q_i = p^\beta_i \; \forall i$, i.e., each q_i is set to the associated marginal distribution of p^β.

2.6 Adaptive importance sampling and Nearest Newton

This subsection shows how to set q to minimize pq distance. It also shows how pq distance provides a tool to perform second order descent over qp distance.

The most straightforward way to estimate the marginal distribution minimizing distance to p^β is via adaptive importance sampling, where the proposal distribution is an earlier version of q itself. In other words, first we write q in terms of the an earlier estimate $\tilde{q}$ as follows:

$$
\begin{aligned}
q_i(x_i) &\propto \int dx'_{-i} \, \frac{e^{-\beta G(x_i, x'_{-i})}}{\tilde{q}_{-i}(x'_{-i})} \, \tilde{q}_{-i}(x'_{-i}) \\
&= \int dx' \, [\frac{e^{-\beta G(x')}}{\tilde{q}(x')}] \, [\tilde{q}(x')\delta(x_i - x'_i)] \\
&= \tilde{q}_i(x_i) \int dx' [\frac{e^{-\beta G(x')}}{\tilde{q}(x')}] \, [\frac{\tilde{q}(x)\delta(x_i - x'_i)}{\tilde{q}_i(x_i)}] \\
&= \tilde{q}_i(x_i) \, E_{\tilde{q}}(\frac{e^{-\beta G}}{\tilde{q}} \mid x_i). \qquad (6)
\end{aligned}
$$

To estimate the expectation value we repeatedly IID sample $\tilde{q}$, getting a set of x values. For each such x we evaluate $e^{-\beta G(x)}/\tilde{q}(x)$. For all of i's possible moves, a, we then set our estimate of $q_i(x_i = a)$ to the average of the subset of those sample values of $e^{-\beta G(x)}/\tilde{q}(x)$ for which $x_i = a$. Variants of this scheme replace the Boltzmann function $e^{-\beta G(x)}$ with a different function that is also biased toward low G values, e.g., $\Theta[K - G(x)]$. See the discussion of iterative focusing below.

Note that the optimal solution for q_i for the pq KL Lagrangian — p_i^β — is independent of the q_{-i}. This is also true for the variants of Monte Carlo sampling with Brouwer updating that are discussed in [7]. This differs from the case for the optimal solution for the qp KL Lagrangian, given in Eq. 2. However in both the adaptive importance sampling scheme for the pq KL Lagrangian and steepest descent for the qp KL Lagrangian, the update rule for q_i depends on the distributions q_{-i}, via the Monte Carlo sampling. See [1].

One potential difficulty with using adaptive importance sampling this way is that it could lead to big jumps in $\mathcal{Q}$, with attendant instability arising from estimation error. An alternative approach for minimizing pq distance is to only use the adaptive importance sampling to provide the information needed to perform steepest descent on the Lagrangians $\mathscr{L}_i^*(q)$. More precisely, the steepest descent direction of that Lagranian for agent i is given by Eq. 3, only now with $u_i(j) = -p_i^\beta(j)/q_i(j)$. The marginal $p_i^\beta(x_i)$ in this direction can be estimated via adaptive importance sampling, just as above. So with such gradient descent we are still using adaptive importance sampling to estimate the marginal of p^β. However now the algorithm only take a small step in a direction determined by that estimate of p_i^β.

Care must be taken when going to second order descent methods. The Hessian of $\mathscr{L}_i^*(q)$ is diagonal and therefore trivial to invert. It is also positivedefinite. However applying a Newton-Raphson step with that Hessian, for example, would result in a new q that doesn't lie on $\mathcal{Q}$. Moreover, the update step is independent of p^β. Such problems are even more accute when trying to descend the Maxent Lagrangian, since coupling between the agents in the multilinear term $E(G)$ makes the associated Hessian non-diagonal.

An alternative approach for second order descent of the Maxent Lagrangian starts by making a quadratic approximation (over the space of all p, not just all q) to the Lagrangian, $\mathscr{L}(p)$ based on the current point p^t. Via Newton's method this specifies a p^{t+1} that minimizes that quadratic approximation. We can then find the product distribution that is nearest (in pq KL distance) to p^{t+1} and move to that product distribution. The resultant update rule for the Maxent Lagrangian is called **Nearest Newton** descent [9]:

$$\begin{aligned}\frac{q_i^{t+1}(j)}{q_i^t(j)} = 1 \quad &- \quad S(q_i^t) - \ln(q_i^t(j)) \\ &- \quad \beta[E_{q^t}(G \mid x_i = j) - E_{q^t}(G)] \end{aligned} \tag{7}$$

where q^t is the current (assumed to be product) distribution. The conditional expectations in Nearest Newton are the same as those in gradient descent. Accordingly, they too can be estimated via Monte Carlo sampling, if need be.

In [13] it is shown that in the continuum time limit, Nearest Newton updating for modifying the probability distribution of any particular agent becomes a variant of replicator dynamics (with the different strategies of replicator dynamics identified with the different possible moves of the agent performing the Nearest Newton update). That paper also shows that that when the terms in that continuum limit version of Nearest Newton are Monte-Carlo estimated, Nearest Newton becomes a version of ficticious play. More precisely, it becomes identical to a "data-aged" continuum time limit of parallel Brouwer updating. The stepsize of the Nearest Newton procedure is identical to the constant in the exponent of the data-aging.

3 Iterative Focusing for pq distance

In common payoff games the ultimate goal is finding $\text{argmin}_x G(x)$, i.e., $\text{argmin}_q E(G)$. More generally, say we have a mechanism design scenario where x is the vector of variables that the designer can set, and the average responses of the human beings to that vector are implicit, being buried (along with the social welfare function) in G. Then the goal of the designer is to find the x minimizing G [42]. This section introduces a class of schemes for minimizing $E_q(G)$, without direct concern for the Maxent Lagrangian.

Given some current distribution, often one can generate distributions that will be more peaked about low $G(x)$ than that current distribution by appropriately transforming it. This suggests a ratchet-like process that iterates a two-step procedure: First one applies such a transformation to the current product distribution, generating a new distributions that in general need not be a product. Then one finds the q that best approximates that transformed version, to get the product distribution for the next iteration. More concretely, the process of **iterative focusing** proceeds as follows:

1) Based on G, choose an associated **focusing operator** $\mathcal{M} : \mathcal{P} \to \mathcal{P}$ such that $E_{\mathcal{M}(p)}(G) < E_p(G)$ for any distribution p. In general $\mathcal{M}(p) \notin \mathcal{Q}$. So we also choose a distance measure $\mathcal{D}(.,.)$ across $\mathcal{P}$, which we will use to get back to $\mathcal{Q}$.

2) Given a product distribution $\tilde{q}$, solve for $\text{argmin}_{q \in \mathcal{Q}} \mathcal{D}(q, \mathcal{M}(\tilde{q}))$. (In general, the finding of that minimizing q may be an extensive multi-step process in its own right.)

3) Set the new $\tilde{q}$ to that solution from (2). Potentially "anneal" $\mathcal{M}$ as well, to tighten the focusing around x with lower values of $G(x)$. Return to (2).

We would like to be able to guarantee that in each pass iterative focusing produces a distribution that is closer to a delta function about $\text{argmin}_x G(x)$. If $\mathcal{M}(\tilde{q})$ were $\in \mathcal{Q}$, so that $\mathcal{D}$ were unnecessary, we would have such a guarantee. However when this is not the case, the distribution output in step (3) $\neq \mathcal{M}(\tilde{q})$, and it may not be superior to the distribution input to step (2). Ultimately whether the distribution produced by step (3) is superior to the one input to step (2) depends on the relation of $\mathcal{D}$, $\mathcal{Q}$, the $\tilde{q}$ input to step (2), and $\mathcal{M}(.)$.

The simplest choice of focusing operator is multiplication by an a nowhere-negative focusing function $F_G(x)$ that increases as $G(x)$ decreases, followed by renormalization:

Proposition 1: Say F_G is integrable and nowhere-negative and $G(x') \leq G(x) \Rightarrow F_G(x') \geq F_G(x)\ \forall x, x'$. Then F_G is a single-valued function of $G(x)$, and in addition the focusing property of step (1) is guaranteed for $\mathcal{M}(p)(x) = \frac{F_G(x)p(x)}{\int dx\ F_G(x)p(x)}$.

Proof: Write $F_G(x) = f(G(x))$ and let y indicate a generic value of $G(x)$. Then the post-focusing distribution over y is

$$P(y) \quad = \quad \int dx \delta(G(x) - y) \frac{f(G(x))p(x)}{\int dx\ f(G(x))p(x)} \quad = \quad \frac{f(y)p(y)}{\int dy\ f(y)p(y)}$$

where $p(y) = \int dx\ p(x)\delta(y - G(x))$. Define the distribution $r(y) \triangleq \frac{f(y)p(y)}{\int dy f(y)p(y)}$. Then

$$\frac{r(y)}{p(y)} \geq 1 \quad \Rightarrow \quad \forall y' \leq y, \frac{r(y')}{p(y')} \geq 1,$$
$$\frac{r(y)}{p(y)} \leq 1 \quad \Rightarrow \quad \forall y' \geq y, \frac{r(y')}{p(y')} \leq 1.$$

Therefore there must be a greatest value a such that $\forall y \leq a, \frac{r(y)}{p(y)} \geq 1$, and a smallest value b such that $\forall y \geq b, \frac{r(y)}{p(y)} \leq 1$. Now by normalization,

$$\int_{-\infty}^{a} dy[r(y) - p(y)] = -\int_{b}^{\infty} dy[r(y) - p(y)],$$

and $r(y) = p(y)$ throughout the range (a, b). Write the change in expected G when multiplying $p(x)$ by $F_G(x)$ and renormalizing as

$$\begin{aligned}\Delta(G) &= \int_{-\infty}^{\infty} dy\; y[r(y) - p(y)] \\ &= \int_{-\infty}^{a} dy\; y[r(y) - p(y)] + \int_{b}^{\infty} dy\; y[r(y) - p(y)].\end{aligned}$$

The first integral is bounded above by $a \int_{-\infty}^{a} dy[r(y) - p(y)]$, and the second is bounded above by $b \int_{b}^{\infty} dy[r(y) - p(y)]$. Combining gives $\Delta(G) \leq 0$. **QED.**

Without loss of generality, we can take $F_G(x)$ to be a probability distribution.

As any example, given any distribution p, the focused distribution

$$\mathcal{M}_{\Theta(K-G(.))}(p)(x) \triangleq \frac{p(x)\Theta[K - G(x)]}{\int dx'\; p(x')\Theta[K - G(x')]}$$

is guaranteed to be more peaked about x with small $G(x)$ than is p. So we can minimize $G(x)$ by iterating the process of finding the product distribution q that best approximates $\mathcal{M}_{\Theta(K-G(.))}(p)$ and then setting $p = q$. In doing this we can also gradually decrease K, to get distributions that are more and more restricted to $\mathrm{argmin}_x G(x)$. This "annealing" is analogous to the an iteration of the process of finding the q that best approximates the Boltzmann distribution for a particular β and then increasing β.

It would be nice to choose a distance measure $\mathcal{D}$ to minimize the chances that the projection back into $\mathcal{Q}$ needed in iterative focusing thwarts the improvement given by applying the focusing operator $\mathcal{M}$. However it's not clear how best to do that. So as an alternative, here we focus on the simplest choices of distance measure, the pq and qp KL distances.

Continuing with our example, for the choice of pq-KL distance as the distance measure $\mathcal{D}$, the q that best approximates $\mathcal{M}_{\Theta(K-G(.))}(p)$ is just the product of the marginal distributions of $\mathcal{M}_{\Theta(K-G(.))}(p)$. So in each pass of the associated iterative focusing algorithm,

$$\begin{aligned}q_i(x_i) &\leftarrow \frac{\int dx'_{-i} q(x'_{-i}, x_i)\Theta[K - G(x_i, x'_{-i})]}{\int dx'\; q(x')\Theta[K - G(x')]} \\ &= \frac{q(G < K, x_i)}{q(G < K)} \\ &= q(x_i \mid G < K) \qquad (8) \\ &\propto q(G < K \mid x_i) q_i(x_i) \qquad (9)\end{aligned}$$

This new q_i can be Monte-Carlo estimated by agent i using only observed G values, in the usual way. So like gradient descent on the Maxent Lagrangian, this update rule is well-suited to a distributed implementation. Indeed, the only term that needs Monte Carlo estimating is $q(G < K \mid x_i)$.

3.1 Iterative focusing and Brouwer updating

One obvious potential problem with this updating rule is that the randomness of the Monte Carlo process might erroneously lead one to set $q_i(x_i = a)$ to 0 even though the x that minimizes G has its i'th component equal to a. Once this happens recovery is impossible; there is no way for $q_i(x_i = a)$ to subsequently increase from 0.

We can avoid this problem if we replace the Heaviside focusing function with a "softened version" like a logistic function with exponent β about K, $\tilde{\Theta}_{\beta,K}(x) \triangleq [1 + e^{\beta(G(x)-K)}]^{-1}$. With this change the update rule becomes

$$q_i(x_i) \leftarrow \frac{E(\tilde{\Theta}_{\beta,K} \mid x_i) q_i(x_i)}{E(\tilde{\Theta}_{\beta,K})}. \tag{10}$$

As another alternative, we can replace the Heaviside function with a Bolzmann distribution with exponent β, getting the update rule

$$q_i(x_i) \leftarrow \frac{E(e^{-\beta G} \mid x_i) q_i(x_i)}{E(e^{-\beta G})}. \tag{11}$$

where all terms on the righthand side are evaluated under the distribution that generated the Monte Carlo samples.

Unlike the update rule of Eq. 9 or Eq. 10, with Eq. 11 we don't need to specify an annealing schedule under which the focusing function changes as the algorithm progresses. The annealing is automatic, due to the fact that multiplying by $e^{-\beta G(x)}$ at each iteration is the same as increasing β by the same constant at each iteration, i.e., due to the fact that the Boltzmann distribution is a Lie group with parameter β.

The update rule of Eq. 11 is similar to that of Eq. 6, only here to form the quantity being averaged one does not divide by $q(x)$. It's also very similar to the **Brouwer update rule** applied to the team game [5, 1, 43]. However in contrast to the potential "thrashing" of parallel Brouwer updating, the update in Eq. 11 is guaranteed to minimize its associated Lagrangian of pq distance to $\mathcal{M}_{\Theta(K-G(.))}(q)$ (assuming no error in estimating the expectation values). On the other hand, in Eq. 11 the Lagrangian is not distance to a fixed distribution, as it is with the Maxent Lagrangian of parallel Brouwer updating and with the update of Eq. 6. Rather the "target distribution" itself changes from one iteration to the next. This is actually the case for all iterative focusing, and is the analogue of the thrashing of parallel Brouwer updating.

3.2 Iterative focusing for qp distance

Rather than pq KL distance, consider using qp KL distance as the error measure for approximating the Boltzmann-focused distribution, $\tilde{q}(x)\frac{e^{-\beta G(x)}}{\int dx' \, \tilde{q}(x')e^{-\beta G(x')}}$. Up to an overall additive constant, that distance is just the Maxent Lagrangian, with one difference: we replace the $S(q)$ term with $-KL(q \;||\; \tilde{q})$. This suggests an iterative algorithm which starts with a distribution $\tilde{q}$, and then searches for the q minimizing a modified version of the Maxent Lagrangian $\mathscr{L}(q)$ in which the $S(q)$ term is replaced by $-KL(q \;||\; \tilde{q})$. When that search terminates one resets $\tilde{q}$ to that minimizing q and starts all over again.

The natural choice for the initial $\tilde{q}$ is the uniform distribution. In this case $-KL(q \;||\; \tilde{q}) = S(q)$, so our KL distance for the first pass of the algorithm reduces to the usual form of the Maxent Lagrangian. Therefore the first stage of the algorithm proceeds to a local minimum q^1 of the Maxent Lagrangian (via Nearest Newton, Brouwer updating, or some such). Once that local minimum is found, there are many schemes to break out of it (see [7]). One of the most obvious is to simply raise (!) β and restart the descent. Such a change to β is equivalent to modifying our current target distribution, $p \triangleq \tilde{q}\frac{e^{-\beta G(x)}}{\int dx' \, \tilde{q}(x')e^{-\beta G(x')}}$, by multiplying it by another Boltzmann distribution (and then renormalizing).

The most natural choice for when to change $\tilde{q}$ to q^1 is after one has gone past all such local minima to a global minimum (or more generally, to what one hopes is a global minimum). However there are many alternatives. For example, one could do the replacement at the first local minimum. As opposed to multiplying the current target p by a Boltzmann distribution, such a replacement of $\tilde{q}$ would make a new target distribution by multiplying q^1 by a Boltzmann distribution. Obvious variants of this scheme weave the resetting of the target p more frequently into the overall process, so that it is updated before a local minimum is found. Indeed, one can even have the target reset after every modification of q, to be the preceding q.

As an example of the foregoing, given the current product distribution $\tilde{q}$, the optimal solution Eq. 2 changes to

$$q_i^{\beta}(x_i) \quad \propto \quad \tilde{q}_i(x_i)e^{-\beta E_{q_{-i}^{\beta}}(G|x_i)}$$

So Brouwer updating is now different from what it is in the conventional Maxent Lagrangian case, with the distribution $\tilde{q}$ serving as a prior probability μ (see Eq. 1). If at each $t+1$ we update $\tilde{q}^{t+1}$ to q^t, and (as in conventional Brouwer) use q^t to estimate the expectation value, the update rule is

$$q^{t+1}(x_i) \quad \propto \quad q_i^t(x_i)e^{-\beta E_{q^t}(G|x_i)}. \tag{12}$$

Just as the parallel, serial, etc., variants of Brouwer updating all have their strengths, so there are reasonable schemes for how to set the updating of the distributions on the righthand side of Eq. 12. For example, one might replace the $q_i^t(x_i)$ term on the righthand side of Eq. 12 with $q_i^{t'}(x_i)$ for some earlier t'. Another possibility is to replace the q^t in the exponential on the righthand side with $q^{t''}$ for some $t'' < t$, and more generally one can vary which of these replacements gets made when.

It is interesting to compare the variant of parallel Brouwer given by Eq. 12 with Nearest Newton applied to the Maxent Lagrangian. If we expand the exponential in Eq. 12 to first order we get

$$q_i^{t+1}(x_i) \propto q_i^t(x_i)[1 - \beta E_{q^t}(G \mid x_i)]. \tag{13}$$

If we now approximate the update of Nearest Newton by removing the ln term (i.e., by taking temperature to 0, with stepsize changed accordingly), we get an update rule almost identical to that of Eq. 13. (The remaining difference is that Nearest Newton normalizes the update to stay in $\mathcal{P}$ by adding a normalizing vector rather than by dividing by a normalizing scalar.) This connection is not too surprising, in light of the fact that in the continuum time limit with data-aging, Nearest Newton and parallel Brouwer updating become identical, with the stepsize of the Nearest Newton identically equal to the data aging-constant in the parallel Brouwer [13].

In addition to Brouwer updating, gradient descent also changes when we use it for iterative focusing of $\tilde{q}$ using qp distance rather than for minimization of the Maxent Lagrangian. The term $u_i(j)$ of Eq. 3 that sets the descent direction becomes

$$\beta E(G \mid x_i = j) + \ln[\frac{q_i(x_i)}{\tilde{q}_i(x_i)}]. \tag{14}$$

So by iteratively focusing $\tilde{q}$ rather than descending the Maxent Lagrangian we penalize q's that differ from $\tilde{q}$. If $\tilde{q}$ is updated frequently, this provides an inertia effect in the dynamics of q, impeding it from changing too fast.[5]

Nearest Newton also changes when used for iterative focusing rather than descending the Maxent Lagrangian. It becomes

$$\begin{aligned} \frac{q_i^{t+1}(j)}{q_i^t(j)} = 1 \quad &+ \quad KL(q_i^t||\tilde{q}_i) - q_i^t(j)\ln[\frac{q_i^t(j)}{\tilde{q}_i(j)}] \\ &- \quad \beta[E_{q^t}(G \mid x_i = j) - E_{q^t}(G)]. \end{aligned} \tag{15}$$

We can similarly implement gradient descent, Nearest Newton, or Brouwer updating for iterative focusing of qp KL distance for other focusing functions besides the Bolzmann function. For example, for the logistic function as the focusing function, we get the same formulas as above, just with $G(x)$ replaced throughout by $\frac{\ln[1+e^{\beta(G(x)-K)}]}{\beta}$.

4 Summary of update rules

In conventional optimization over Euclidean spaces, one can use many different algorithms, including gradient descent, conjugate gradient, Newton's method,

[5] In the limit where $\tilde{q}$ is always set to the q just before the current one, our gradient descent becomes a second order dynamical equation. Iterative focusing based on several previous distributions, not just the single distribution $\tilde{q}$, leads to higher-order dynamics.

quasi-Newton, simulated annealing, genetic algorithms, etc. As proven in [44], it is theoretically impossible for any one of these problems to be superior to the others overall. Rather which algorithm one should use depends on the nature of the function being optimized, the expense of evaluating various kinds of information during the optimization, etc.

Similarly, in the preceding discussion many different PC update rules were presented, with the ultimate choice of which rule to use depending on the nature of the function being optimized, the expense of evaluating various kinds of information during the optimization, etc. This section compares some of those update rules.

4.1 Multiplicative updating

All the update rules described above can be written as multiplicative updating. The following is a list of the update ratios $r_{q,i}(x_i) \triangleq q_i^{t+1}(x_i)/q_i^t(x_i)$ of some of those rules. In all of these F_G is a probability distribution over x that never increases between two x's if G does (e.g., a Boltzmann distribution in $G(x)$). In addition const is always a scalar that ensures the new distribution is properly normalized and α is a stepsize.[6]

Gradient descent of qp distance to F_G:

$$1 - \alpha[\frac{E_{q^t}(\ln[F_G] \mid x_i) + \ln(q_i^t(x_i))}{q_i^t(x_i)}] - \frac{\text{const}}{q_i^t(x_i)} \tag{16}$$

Nearest Newton descent of qp distance to F_G:

$$1 - \alpha[E_{q^t}(\ln[F_G] \mid x_i) + \ln(q_i^t(x_i))] - \text{const} \tag{17}$$

Brouwer updating for qp distance to F_G:

$$\text{const} \times \frac{e^{E_{q^t}(\ln[F_G] \mid x_i)}}{q_i^t(x_i)} \tag{18}$$

Importance sampling minimization of pq distance to $F_G(x)$:

$$\text{const} \times E_{q^t}(\frac{F_G}{q^t} \mid x_i) \tag{19}$$

Iterative focusing of $\tilde{q}$ with focusing function $F_G(x)$ using qp distance and gradient descent:

$$1 - \alpha\{\frac{E_{q^t}(\ln[F_G] \mid x_i) + \ln[\frac{q_i^t(x_i)}{\tilde{q}_i(x_i)}]}{q^t(x_i)}\} - \frac{\text{const}}{q^t(x_i)} \tag{20}$$

[6] As a practical matter, both Nearest Newton and gradient-based updating have to be modified in a particular step if their step size is large enough so that they would otherwise take one off the unit simplex. This changes the update ratio for that step. See [9].

Iterative focusing of $\tilde{q}$ with focusing function $F_G(x)$ using qp distance and Nearest Newton:

$$1 - \alpha\{E_{q^t}(\ln[F_G] \mid x_i) + \ln[\frac{q_i^t(x_i)}{\tilde{q}_i(x_i)}]\} - \text{const} \tag{21}$$

Iterative focusing of $\tilde{q}$ with focusing function $F_G(x)$ using qp distance and Brouwer updating:

$$\text{const} \times e^{E_{q^t}(\ln[F_G] \mid x_i)} \times \frac{\tilde{q}(x_i)}{q_i^t(x_i)} \tag{22}$$

Iterative focusing of $\tilde{q}$ with focusing function $F_G(x)$ using pq distance:

$$\text{const} \times E_{\tilde{q}}(F_G \mid x_i) \times \frac{\tilde{q}(x_i)}{q_i^t(x_i)} \tag{23}$$

Note that some of these update ratios are themselves proper probability distributions, e.g., the Nearest Newton update ratio.

All of these update rules are invariant under rescaling of $F_G(x)$, i.e., multiplication of $F_G(x)$ by a constant (the term const changes to compensate for the new scale). When $F_G(x)$ is an exponential (i.e., Boltzmann function) of $G(x)$, such a transformation of $F_G(x)$ is equivalent to adding a constant to $G(x)$. However for the other choices of $F_G(x)$ mentioned above (e.g., $\Theta(K - G(x))$), there is no simple correspondence between this invariance of the update rule and a change to $G(x)$. In particular, for those other F_G, rescaling of $F_G(x)$ does not correspond to adding a constant to $G(x)$. Accordingly, unlike the case for exponential $F_G(x)$, those non-exponential $F_G(x)$ do not give rise to update rules that are invariant under addition of a constant to $G(x)$. The exponential $F_G(x)$ has the other nice property that rescaling $G(x)$ is equivalent to just translating the exponent constant β, i.e., to tightening $F_G(x)$ about x with low $G(x)$. Just as other $F_G(x)$ do not have nice behavior under addition of a constant to $G(x)$, they also do not have this nice character under rescaling.

Simple modifications to non-exponential F_G typically allows them to share these nice characteristics. These modifications involve dynamically replacing constants in those F_G, in particular replacing constants that might otherwise be annealed according to a pre-fixed schedule. Typically in the place of such constants one uses explicit functions of $\tilde{q}$. For example, we can replace $\Theta(K - G(x))$ with $\Theta(E_{\tilde{q}}(G) - G(x))$, or with $\Theta(\pi(\epsilon, \tilde{q}, G) - G(x))$, where $\pi(\epsilon, q, G)$ is the value of $G(x)$ in the best ϵ percentile under distribution q. For either of these replacements $F_G(x)$ is invariant under both translation and scaling of $G(x)$.

Say one has a set of multiple objective functions / constraints $\{G_i\}$ rather than just a single, unconstrained function G.[7] Some of the update rules presented above can be modified to simultaneously improve all of those functions.

[7] For current purposes, we can cast constraints over x as objective functions, so that the expectation of those functions equals 0 iff the underlying distribution has its support restricted to x meeting the constraints.

One does this by replacing "$F_G(x)$" throughout the update rule with some function $F_{\{G_i\}}(x)$. That function is designed so that its global maxima contain a Pareto optimal x (according to the $\{G_i\}$). In addition it cannot increase (i.e., improve) in going from one x to another if any of the $F_{G_i}(x)$ increase in that transition.[8] A simple example, applicable for update rules using *pq* distance, is $F_{\{G_i\}}(x) = \prod_i \Theta(K_i - G_i(x))$. In practice of course, the choice of the function $F_{\{G_i\}}(x)$ and the update rule one is modifying will have a crucial effect on how prone the algorithm is to getting caught at local critical points.

These update rules can be broadly grouped into two distinct sets based on what guarantees they have. Say we can evaluate every term in each update rule in closed form, or alternatively that our Monte Carlo estimate is exact. Then there is a q-independent target distribution, p^*, and a distance measure $\mathcal{D}$, such that each update of q in the serial Brouwer version of Eq. 18 is guaranteed not to increase $\mathcal{D}(q, p^*)$. (In this case p^* is the Boltzmann distribution over values of F_G, and $\mathcal{D}$ is *qp* distance.) The same is true for for adaptive importance sampling minimization of *pq* distance, Eq. 19. (Again p^* is the Boltzmann distribution in F_G.) For small enough step size, we also have this guarantee for gradient descent of *qp* distance and Nearest Newton. None of the other update rules have such guarantees.

Finally, it is worth re-emphasizing that for the update rule of adaptive importance sampling minimization of *pq* distance, at equilibrium each q_i is independent of the distributions q_{-i}. (It's just the marginal of the Boltzmann distribution p^β.) As mentioned previously, the same property holds for the variants of the Monte Carlo process discussed in the section on modifying the Monte Carlo process for Brouwer updating discussed in [7]. This means one can use samples from all the previous Monte Carlo blocks with impunity. You don't have to worry that the samples of those earlier blocks were formed under a different q_{-i}, and therefore would lead to a different update from the one appropriate for the current Monte Carlo block. In addition each of these two algorithms have a single equilibrium, given by the Boltzmann distribution p^β. In this, they have no local minima problems. None of the other update rules have such a set of guarantees.

References

[1] D. H. Wolpert, "Factoring a canonical ensemble," 2003, cond-mat/0307630.

[2] ——, "Information theory — the bridge connecting bounded rational game theory and statistical physics," in *Complex Engineering Systems*, A. M. D. Braha and Y. Bar-Yam, Eds., 2004.

[8]Such a function is only a partial ordering over x. In particular, consider a change in x which improves some G_i while all the others get worse. Often we can follow that change with another in which the improving and worsening G_i flip roles, so that in aggregate all the $G_i(x)$ have shrunk, even though $F_{\{G_i\}}(x)$ cannot have decreased.

[3] W. Macready, S. Bieniawski, and D. Wolpert, “Adaptive multi-agent systems for constrained optimization,” 2004, technical report IC-04-123.

[4] C. F. Lee and D. H. Wolpert, “Product distribution theory for control of multi-agent systems,” in *Proceedings of AAMAS 04*, 2004.

[5] S. Bieniawski and D. H. Wolpert, “Adaptive, distributed control of constrained multi-agent systems,” in *Proceedings of AAMAS04*, 2004.

[6] S. Bieniawski, D. H. Wolpert, and I. Kroo, “Discrete, continuous, and constrained optimization using collectives,” in *Proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York*, 2004, in press.

[7] D. H. Wolpert, “Finding bounded rational equilibria part ii: Alternative lagrangians and uncountable move spaces,” 2004, http://xxx...

[8] S. Airiau and D. H. Wolpert, “Product distribution theory and semi-coordinate transformations,” 2004, submitted to AAMAS 04.

[9] D. H. Wolpert and S. Bieniawski, “Distributed control by lagrangian steepest descent,” in *Proceedings of CDC04*, 2004.

[10] ——, “Adaptive distributed control: beyond single-instant categorical variables,” in *Proceedings of MSRAS04*, A. S. et al, Ed. Springer Verlag, 2004.

[11] N. Antoine, S. Bieniawski, I. Kroo, and D. H. Wolpert, “Fleet assignment using collective intelligence,” in *Proceedings of 42nd Aerospace Sciences Meeting*, 2004, aIAA-2004-0622.

[12] D. H. Wolpert and C. F. Lee, “Adaptive metropolis hastings sampling using product distributions,” in *Proceedings of ICCS 04*, 2004.

[13] D. H. Wolpert, “What information theory says about best response, binding contracts, and collective intelligence,” in *Proceedings of WEHIA04*, A. N. et al, Ed. Springer Verlag, 2004.

[14] D. H. Wolpert, K. Tumer, and J. Frank, “Using collective intelligence to route internet traffic,” in *Advances in Neural Information Processing Systems - 11*. MIT Press, 1999, pp. 952–958.

[15] D. H. Wolpert and K. Tumer, “Optimal payoff functions for members of collectives,” *Advances in Complex Systems*, vol. 4, no. 2/3, pp. 265–279, 2001.

[16] ——, “Collective intelligence, data routing and braess’ paradox,” *Journal of Artificial Intelligence Research*, 2002.

[17] D. H. Wolpert, “Theory of collective intelligence,” in *Collectives and the Design of Complex Systems*, K. Tumer and D. H. Wolpert, Eds. New York: Springer, 2003.

[18] D. Fudenberg and J. Tirole, *Game Theory.* Cambridge, MA: MIT Press, 1991.

[19] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory.* Philadelphia, PA: Siam, 1999, second Edition.

[20] M. Osborne and A. Rubenstein, *A Course in Game Theory.* Cambridge, MA: MIT Press, 1994.

[21] R. Aumann and S. Hart, *Handbook of Game Theory with Economic Applications.* North-Holland Press, 1992.

[22] R. Axelrod, *The Evolution of Cooperation.* NY: Basic Books, 1984.

[23] T. Sandholm and V. R. Lesser, "Coalitions among computationally bounded agents," *Artificial Intelligence*, vol. 94, pp. 99–137, 1997.

[24] A. Neyman, "Bounded complexity justifies cooperation in the finitely repeated prisoner's dilemma," *Economics Letters*, vol. 19, pp. 227–230, 1985.

[25] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games.* Cambridge, MA: MIT Press, 1998.

[26] W. B. Arthur, "Complexity in economic theory: Inductive reasoning and bounded rationality," *The American Economic Review*, vol. 84, no. 2, pp. 406–411, May 1994.

[27] C. Boutilier, Y. Shoham, and M. P. Wellman, "Editorial: Economic principles of multi-agent systems," *Artificial Intelligence Journal*, vol. 94, pp. 1–6, 1997.

[28] N. I. Al-Najjar and R. Smorodinsky, "Large nonanonymous repeated games," *Game and Economic Behavior*, vol. 37, no. 26-39, 2001.

[29] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and Uncertainty*, vol. 5, pp. 297–323, 1992.

[30] D. Kahneman, "A psychological perspective on economics," *American Economic Review (Proceedings)*, vol. 93:2, pp. 162–168, 2003.

[31] T. Cover and J. Thomas, *Elements of Information Theory.* New York: Wiley-Interscience, 1991.

[32] D. Mackay, *Information theory, inference, and learning algorithms.* Cambridge University Press, 2003.

[33] E. T. Jaynes and G. L. Bretthorst, *Probability Theory : The Logic of Science.* Cambridge University Press, 2003.

[34] E. T. Jaynes, "Information theory and statistical mechanics," *Physical Review*, vol. 106, p. 620, 1957.

[35] W. Macready and D. H. Wolpert, "Distributed constrained optimization with semi-coordinate transformations," 2004, submitted.

[36] ——, "Distributed optimization," in *Proceedings of ICCS 04*, 2004.

[37] J. Shamma and G. Arslan, "Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria," 2004, submitted.

[38] D. Fudenberg and D. Kreps, "Learning mixed equilibria," *Game and Economic Behavior*, vol. 5, pp. 320–367, 1993.

[39] D. Fudenberg and D. K. Levine, "Steady state learning and Nash equilibrium," *Econometrica*, vol. 61, no. 3, pp. 547–573, 1993.

[40] D. H. Wolpert and W. Macready, "Self-dissimilarity as a high dimensional complexity measure," in *Proceedings of the International Conference on Complex Systems, 2004*, 2004, in press.

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd ed.)*. Wiley and Sons, 2000.

[42] D. Wolpert and K. Tumer, "Beyond mechanism design," in *International Congress of Mathematicians 2002 Proceedings*, H. G. et al., Ed. Qingdao Publishing, 2002.

[43] S. Bieniawski and D. H. Wolpert, "Using product distributions for distributed optimization," in *Proceedings of ICCS 04*, 2004.

[44] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997, best Paper Award.