Distributed Constrained Optimization

William Macready
NASA Ames Research Center,
MailStop 269-4,
Moffett Field, CA, 94035
wgm@email.arc.nasa.gov

David Wolpert
NASA Ames Research Center,
MailStop 269-4,
Moffett Field, CA, 94035
dhw@email.arc.nasa.gov

Abstract

We demonstrate a new framework for analyzing and controlling distributed systems, by solving constrained optimization problems with an algorithm based on that framework. The framework is an information-theoretic extension of conventional full-rationality game theory to allow bounded rational agents. The associated optimization algorithm is a game in which agents control the variables of the optimization problem. They do this by jointly minimizing a Lagrangian of (the probability distribution of) their joint state. The updating of the Lagrange parameters in that Lagrangian is a form of automated annealing, one that focuses the multi-agent system on the optimal pure strategy. We present computer experiments for the k-sat constraint satisfaction problem and for unconstrained minimization of NK functions.

1 Introduction

Recently, a new framework called probability collectives (PC) has been designed for for analyzing, optimizing and controlling distributed systems [1, 2, 3]. One goal of this work is to develop algorithms by which a distributed collection of agents can be coordinated to perform desired tasks. Here we consider constrained optimization tasks where a distributed solution may be desired either because the variables and constraints between variables are spread across many agents (as in distributed design or supply chain application), or simply because it is advantageous to £nd a solution method which can be easily parallelized so that large problem instances may be solved.

The natural way to map a multi-agent collective onto an optimization task is to assign an agent to each variable x_i in the problem. If the domain of the ith variable is $\mathcal{X}_i$ then the ith agent is responsible for selecting a value for x_i from $\mathcal{X}_i$. The $|\mathcal{X}_i|$ possible selections become the possible moves of the agent. The joint set of n variables (agents) describing the system is indicated as $\mathbf{x} = [x_1, \cdots, x_n] \in \mathcal{X}$ with $\mathcal{X} \equiv \mathcal{X}_1 \times \cdots \times \mathcal{X}_n$.[1] Unlike many optimization methods the variables are set through the determination of a probability distribution q over $\mathcal{X}$. A Lagrangian function, $\mathcal{L} : \mathcal{Q} \mapsto \mathbb{R}$, is de£ned and minimized to determine the optimal q from within a set $\mathcal{Q}$ of possible probability distributions over $\mathcal{X}$.

Optimizing over $\mathcal{Q}$ rather than $\mathcal{X}$ simpli£es optimization tasks over discrete variables.

[1] Vectors are indicated in bold font and scalars are in regular font.

Since $q \in \mathcal{Q}$ is a vector in a Euclidean space, the search can be done with continuous techniques like gradient descent or Newton's method – even if $\mathcal{X}$ is a categorical, £nite space. Our approach differs from most stochastic optimization algorithms like simulated annealing. Typically, those algorithms use samples from probability distributions (e.g. Boltzmann distribution in the case of simulated annealing) to help guide search for points $\mathbf{x}$ optimizing an objective function $G(\mathbf{x})$. In contrast, while still utilizing probability distributions (we too will use a Boltzmann distribution) we search over distributions directly.

A strength of the PC framework is the connections it makes to relate disciplines to one another. For example, it can be motivated by using information theory to relate bounded rational game theory to statistical physics [1, 2]. In a noncooperative game the agents are statistically independent at any stage of the game, with each agent i choosing its move x_i by sampling its probability distribution (mixed strategy) at that instant, $q_i(x_i)$; the distribution of the joint-moves is then a product distribution. Inter-agent coupling occurs indirectly, across time, via the updating of the $\{q_i\}_{i=1}^n$ at the end of each stage. Information theory shows that the bounded rational equilibrium of the game is the q optimizing an associated Lagrangian $\mathcal{L}(q)$. Applying these ideas to distributed optimization we assign an agent to each variable x_i where the setting of $x_i \in \mathcal{X}_i$ is determined by sampling from $q_i(x_i)$. As the agents strategies are independent at any given time, all the variables may be updated in parallel. The Lagrangian which couples the q_i depends on the objective $G(\mathbf{x})$ being minimized. Coordinate descent on $\mathcal{L}(q)$ determines the update of q.

For some games an appropriate Lagrangian is the Kullback-Leibler (KL) distance[2] to a known distribution p: $D(q\|p) \equiv \sum_{\mathbf{x}} q(\mathbf{x}) \ln\big(q(\mathbf{x})/p(\mathbf{x})\big)$ [4]. Typically, p is one of the ensembles of statistical physics. In the work presented here we will use the canonical ensemble governed by the Boltzmann distribution $p(\mathbf{x}) \propto \exp[-G(\mathbf{x})/T]$ which arises as the maximum entropy distribution resulting from a speci£cation of the average payoff (which is to be minimized) shared by all agents. The KL distance $D(q,p)$ to the Boltzmann is proportional to the Helmholtz free energy of statistical physics so that the optimizer of $\mathcal{L}(q)$ may be interpreted as the distribution that minimizes the expected value of G, subject to any provided constraints and to an overall entropy value that sets the rationalities of the agents. For $\mathcal{Q}$ being the set of product distributions, the bounded rational equilibrium of the game is then a mean-£eld approximation to p.

The game theoretic motivations considered above suggest that $\mathcal{Q}$ should often be the set of product distributions over $\mathcal{X}$. This choice allows for a highly parallel algorithm, but other concerns may dictate different $\mathcal{Q}$. In many optimization tasks we seek multiple solutions. In Constraint Satisfaction Problems (CSPs) [**?**] in particular, the goal is to identify feasible solutions which satisfy a set of constraints. For small problem instances exhaustive enumeration techniques like branch-and-bound are typically used to identify all feasible solutions of a CSP, or to show that none exist. In cases like these, where we desire multiple solutions, a product distribution may be a poor choice. A converged product distribution $q(\mathbf{x}) = \prod_{i=1}^n \delta(x_i - x_i^*)$ can only represent the single solution $\mathbf{x}^*$. If we desire many solutions we might descend on the Lagrangian beginning from different starting points (i.e. different initial q), but there is no guarantee that multiple runs will each generate different solutions. The PC framework offers a simple solution to this problem – extend $\mathcal{Q}$ to construct a single game where we obtain multiple distinct solutions at once. The approach is to de£ne a space $\mathcal{X}'$ so that a product distribution over $\mathcal{X}'$ corresponds to a coupled distribution across $\mathcal{X}$. We consider a $\mathcal{X}'$ that results in a mixture of M product distributions $q(\mathbf{x}) = \sum_m q_0(m) q^m(\mathbf{x})$ [**?**] over $\mathcal{X}$. This allows for the determination of M solutions at once with a Lagrangian providing a term which "pushes" the separate products $q^m(\mathbf{x})$ apart to favor distinct solutions.

[2]The KL distance $D(q\|p)$ between two probability distributions q and p is not a metric but is non-negative and equal to zero only when $q = p$.

We begin in section 2 by elaborating a Lagrangian for single product distribution models, and consider two methods to minimize this Lagrangian in section 3. Section 4 extends product Lagrangians to allow for mixture models, and shows how mixture models may be seen as a product distributions over a different space. Experimental validation is presented for the k-satis£ability CSP problem (section 6.1) and the NK (section 6.2) optimization problems.

2 The Lagrangian for Product Distributions

We begin with product distributions. To specify the Lagrangian we £x the distribution $p(\mathbf{x})$ we wish to approximate (in KL distance). If the objective function we wish to minimize is $G(\mathbf{x})$ (i.e., G is the negative of the utility shared by the bounded rational agents), then it is natural to consider the T-parameterized Boltzmann distribution $p(\mathbf{x}) = \exp[-G(\mathbf{x})/T]/Z(T)$. At low T — high rationalities — this distribution is concentrated on $\mathbf{x}$ having low G values. For a given q the KL distance to p is proportional to

$$\mathcal{L}(q) = \mathbb{E}_q(G) - TS(q) \tag{1}$$

where $\mathbb{E}_q(G) \equiv \sum_x q(x)G(x)$, and $S(q) \equiv -\sum_{\mathbf{x}} q(\mathbf{x}) \ln q(\mathbf{x})$ is the entropy of q. For q's which are product distributions $S(q) = \sum_i S(q_i)$ where $S(q_i) = -\sum_{x_i} q_i(x_i) \ln q_i(x_i)$. The £rst term in $\mathcal{L}$ is minimized by a perfectly rational player, i.e. by a player who concentrates all probability on the best move. The second term is minimized by a perfectly irrational player, i.e., by a perfectly uniform mixed strategy q_i. So T speci£es the balance between the rational and irrational behavior of the player. In particular, for $T \to 0$, by minimizing the Lagrangian we recover the Nash equilibria of the game. From a statistical physics perspective where T is recognized as the temperature the Lagrangian is simply the Gibbs free energy of statistical physics for the Hamiltonian G.

Since we are interested in problems with constraints, we write

$$G(\mathbf{x}) = O(\mathbf{x}) + \sum_{a=1}^{C} \lambda_a c_a(\mathbf{x})$$

where O is an objective to be minimized, and the c_a are a set of C inequality constraint functions that are required to be less than or equal to zero. The λ_a are the Lagrange multipliers that are used to enforce the constraints. In CSP's we take $O(\mathbf{x}) = 0$.

3 Minimizing the Product Lagrangian

$\mathcal{L}$ must be minimized subject to the imposed constraints $\{c_a\}_{a=1}^{C}$ on $\mathbf{x}$ to determine the $\sum_{i=1}^{n} |\mathcal{X}_i|$ continuous variables $\{q_i(x_i)\}_{i=1}^{n}$ and the C Lagrange multipliers $\{\lambda_a\}_{a=1}^{C}$. As the optimization variables de£ne a probability we have additional constraints on the optimization variables: $0 \leq q_i(x_i) \leq 1$ for all i and x_i, and $\sum_{x_i} q_i(x_i) = 1$ for all i. We consider two approaches to determining the q_i and note the gradient descent update rule for the Lagrange multipliers λ_a appearing in G.

3.1 Brouwer Updating

Equating the gradient of $\mathcal{L}$ with respect to $q_i(x_i)$ at step t to zero yields the Boltzmann update rule

$$q_i^{t+1}(x_i) \propto \exp\left[-\mathbb{E}_{q_{\setminus i}^t}(G|x_i)/T\right] \tag{2}$$

where the private "utilities" for the ith agent is

$$\mathbb{E}_{q_{\setminus i}}(G|x_i) = \sum_{\mathbf{x}_{\setminus i}} q_{\setminus i}(\mathbf{x}_{\setminus i}) G(x_i, \mathbf{x}_{\setminus i}) \tag{3}$$

with $\mathbf{x}_{\setminus i} = [x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n]$ and $q_{\setminus i}(\mathbf{x}_{\setminus i}) = \prod_{j=1|j\neq i}^{n} q_j(x_j)$. This local measure is the expected payoff to agent i as measured by the distribution $q_{\setminus i}$ across the moves of all other agents when i plays move x_i. In the usual way, we may update the agents one at a time or in parallel with this rule. However, in the parallel updating case we have no guarantees that the iterations will converge.

3.2 Nearest-Newton Updating

The special structure of the Lagrangian allows for the simple inclusion of second order information for fast Newton-like descent. This nearest-Newton updating rule begins from from the observation that the Lagrangian, $\mathbb{E}_\pi(G) - TS(\pi)$, for an unrestricted probability distribution[3] π is a convex function of π with a diagonal Hessian. One way to exploit this fact is as follows: from the current q^t make an unrestricted Newton step which will result in a distribution π^{t+1} that is typically not in $\mathcal{Q}$, and then £nd the $q^{t+1} \in \mathcal{Q}$ that is nearest to π^{t+1}.

As the Hessian $\partial^2\mathcal{L}/\partial\pi(\mathbf{x})\partial\pi(\mathbf{x}')$ is diagonal it is simply inverted, and the Newton update for π^t is

$$\pi^{t+1}(\mathbf{x}) = \pi^t(\mathbf{x}) - \alpha^t\pi^t(\mathbf{x})\left[\frac{G(\mathbf{x}) - \mathbb{E}_{\pi^t}(G)}{T} + S(\pi^t) + \ln\pi^t(\mathbf{x})\right]$$

which is normalized if π^t is normalized and where α^t is a step size. As π^t will typically not belong to $\mathcal{Q}$ we £nd the product distribution nearest to π^{t+1} by minimizing the KL distance $D(\pi^{t+1}\|q)$ with respect to q. The result is that $q_i(x_i) = \pi_i^{t+1}(x_i)$, i.e. q_i is the corresponding marginal of π^{t+1}. Thus, assuming that π^t is also a product distribution (as it must be according to our product assumption) then the update rule for $q_i(x_i)$ is

$$q_i^{t+1}(x_i) = q_i^t(x_i) - \alpha^t q_i^t(x_i)\left[\frac{\mathbb{E}_{q_{\setminus i}^t}(G|x_i) - \mathbb{E}_{q^t}(G)}{T} + S(q_i) + \ln q_i(x_i)\right]. \qquad (4)$$

This update maintains the normalization of q_i, but may make one or more $q_i(x_i)$ greater than 1 or less than 0. In such cases we set q^t to be valid probability distribution nearest (in Euclidean distance) to the suggested Newton update.

3.3 Estimation of $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$

Both update rules Eqs. (2) and (4) require $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$ de£ned in Eq. (3). Depending on the problem at hand this expectation may be evaluated in closed form, or it may be estimated by Monte Carlo sampling. For the problems considered here the expectation may be ef£ciently calculated in closed form, but for completeness we present a Monte Carlo approach that minimizes the need for excessive sampling.

All that is important in the updates for $q_i(x_i)$ are the differences $\mathbb{E}_{q_{\setminus i}^t}(G|x_i) - \mathbb{E}_{q_{\setminus i}^t}(G|x_i')$ for pairs of distinct moves x_i and x_i'. The magnitudes of the $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$ are absorbed into the normalization of q_i. Consequently, rather than the use the sample average of $G(\mathbf{x})$ we can use the sample average of $g_i(\mathbf{x}) = G(\mathbf{x}) - h_i(\mathbf{x}_{\setminus i})$ which will leave the differences unaffected. The function $h_i(\mathbf{x}_{\setminus i})$ can be chosen so that the Monte Carlo estimate has both low bias (with respect to estimating $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$) and low variance [5]. Intuitively, the bias re¤ects the alignment between the private utilities g_i, and the world utility G. At zero bias, reducing private utility necessarily reduces world utility. Variance instead re¤ects how

[3] I.e. we do not insist that π is a product or have any particular form, only that all $0 \leq \pi(\mathbf{x}) \leq 1$ and $\sum_{\mathbf{x}} \pi(\mathbf{x}) = 1$.

much the utility depends on the agent's own move rather than those of the other agents. With low variance, the agents can perform the individual optimizations accurately with minimal Monte-Carlo sampling.

The *Aristocrat Utility* (AU) is the estimator, out of all those guaranteed to have zero bias, that has minimal variance:

$$g_i^{AU}(x_i, \mathbf{x}_{\setminus i}) = G(x_i, \mathbf{x}_{\setminus i}) - \sum_{x_i'} \frac{N_{x_i'}^{-1}}{\sum_{x_i''} N_{x_i''}^{-1}} G(x_i', \mathbf{x}_{\setminus i}) \tag{5}$$

where N_{x_i} is the number of times that agent i makes move x_i in the most recent set of Monte Carlo samples. Unfortunately, evaluation of the AU private utility can be expensive as it requires numerous calls to G. A cheaper alternative can be derived by noting that the weighting factor $N_{x_i'}^{-1} / \sum_{x_i''} N_{x_i''}^{-1}$ is largest for those x_i which occur infrequently, i.e. that have low $q_i(x_i)$. This observation leads to the *Wonderful Life Utility* (WLU), which is an approximation to AU that also has zero bias:

$$g_i^{WLU}(x_i, \mathbf{x}_{\setminus i}) = G(x_i, \mathbf{x}_{\setminus i)}) - G(x_i^{\text{clamp}}, \mathbf{x}_{\setminus i}). \tag{6}$$

In the above, $x_i^{\text{clamp}} = \arg\min_{x_i} q_i(x_i)$ is agent i's lowest probability move [1, 3].

Further computational speedups in the expectation may be obtained by smoothing the Monte Carlo estimates by ageing the data. If the updates to q are not changing rapidly (as will be the case for small step sizes α^t or when q nears a local minimum), then the expectations $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$ will not vary greatly with t and we may use the estimate at $\mathbb{E}_{q_{\setminus i}^t}(G|x_i) = (1-\gamma)\sum_{\mathbf{x}}' g_i(\mathbf{x}) + \gamma \mathbb{E}_{q_{\setminus i}^{t-1}}(G|x_i)$ where $\sum_{\mathbf{x}}'$ sums over samples whose ith component is x_i, and where γ is an aging parameter.

In this paper we examine problems for which the required expectations $\mathbb{E}_{q_{\setminus i}^t}(G|x_i)$ may be obtained in closed form so that there is no need for Monte Carlo approximations. However, comparisons of the different utility functions and the effects of the ageing parameter may be found in **[REF TO STEFAN B paper]**

3.4 Updating Lagrange Multipliers

In order to satisfy the imposed optimization constraints $\{c_a\}$ we must also update the Lagrange multipliers. To minimize communication between agents this is done in the simplest possible way – by gradient descent. Taking the partial derivatives with respect to λ_a gives the update rule

$$\lambda_a^{t+1} = \lambda_a^t + \alpha_\lambda^t \mathbb{E}_{q^*}\big(c_a(\mathbf{x})\big) \tag{7}$$

where α_λ^t is a step size and q^* is the minimizer of $\mathcal{L}$ determined as above at the old settings, λ^t, of the multipliers.

3.5 Agent Communication

All agents (variables) sample moves (variable settings) independently, and coupling occurs only in the updates of the q_i. As we have seen this update (even to second order) for agent i depends only on the conditional expectations $\mathbb{E}_{q_{\setminus i}}(G|x_i)$ where $q_{\setminus i}$ describes the strategies used by the other agents. Thus, if we are using Monte Carlo, then the only information which needs to be communicated to each agent is the G values upon which the estimate will be based. Using these values each agent independently updates its strategy (its q_i) in a way which collectively is guaranteed to lower the Lagrangian. If the expectation is evaluated analytically, the ith agent needs the q_j distributions for each of the the j agents involved in factors with i. For objectives (e.g. the problems considered here) which consists

of a sum of local interactions each of which individually involves only a small subset of the variables, the number of agents that i needs to communicate with may be much smaller than n.

4 Mixture Distributions

We have described how a Lagrangian which measures the distance of a product (mean £eld) distribution to a Boltzmann distribution may be de£ned and minimized in a distributed fashion. We now extend these results to mixtures of product distributions in order to represent multiple solutions. However, before doing so we demonstrate that a mixture distribution may be viewed as a product distribution over a different set of variables.

4.1 Coordinate Transformations – mixtures as products

Let $\mathbf{f} \in \mathcal{F}$ indicate the new set of variables in a space of dimension $d_{\mathcal{F}}$. A product distribution assumption over $\mathcal{F}$ (where $d_{\mathcal{F}} > n$), and an appropriately chosen mapping $\boldsymbol{\zeta} : \mathcal{F} \mapsto \mathcal{X}$ induces a mixture distribution over $\mathcal{X}$.

Consider an M component mixture distribution over n variables: $\sum_{m=1}^{M} q^0(m) q^m(\mathbf{x})$ with $\sum_{m=1}^{M} q^0(m) = 1$ and $q^m(\mathbf{x}) = \prod_{i=1}^{n} q_i^m(x_i)$. We can write this as a product distribution space of dimension $d_{\mathcal{F}} = 1 + Mn$ where the £rst dimension (indicated as $f^0 \in [1, M]$) labels the mixtures, and where the remaining Mn dimensions (indicated as $f_i^m \in \mathcal{X}_i$) correspond to each of the original n dimensions for each of the M mixtures. The $\mathcal{F}$-space product distribution takes the form $q_{\mathcal{F}}(\mathbf{f}) = q^0(f^0) \prod_{m=1}^{M} q^m(\mathbf{f}^m)$ with $q^m(\mathbf{f}^m) = \prod_{i=1}^{N} q_i^m(f_i^m)$ for $\mathbf{f} = [f^0, \mathbf{f}^1, \cdots, \mathbf{f}^M]$ and $\mathbf{f}^m = [f_1^m, \cdots, f_N^m]$. The density in $\mathcal{F}$ and $\mathcal{X}$ are related as usual by $q(\mathbf{x}) = \sum_{\mathbf{f}} q_{\mathcal{F}}(\mathbf{f}) \delta\big(\mathbf{x} - \boldsymbol{\zeta}(\mathbf{f})\big)$ for some vector-valued mapping $\boldsymbol{\zeta} : \mathcal{F} \mapsto \mathcal{X}$, and with the delta function of vectors being understood component-wise. If we label the components of $\boldsymbol{\zeta}$ so that $x_i = \zeta_i(f^0, \mathbf{f}^1, \cdots, \mathbf{f}^M) = f_i^{f^0}$ we £nd

$$
\begin{aligned}
q(\mathbf{x}) &= \sum_{f^0} q^0(f^0) \sum_{\mathbf{f}^1, \cdots, \mathbf{f}^M} \prod_m q^m(\mathbf{f}^m) \prod_i \delta\big(x_i - \zeta_i(f^0, \mathbf{f}^1, \cdots, \mathbf{f}^M)\big) \\
&= \sum_{f^0} q^0(f^0) \sum_{\mathbf{f}^1, \cdots, \mathbf{f}^M} \prod_m q^m(\mathbf{f}^m) \prod_i \delta\big(x_i - f_i^{f^0}\big) \\
&= \sum_{f^0} q^0(f^0) \sum_{\mathbf{f}^{f^0}} q^{f^0}(\mathbf{f}^{f^0}) \prod_i \delta\big(x_i - f_i^{f^0}\big) \\
&= \sum_{f^0} q^0(f^0) q^{f^0}(\mathbf{x})
\end{aligned}
$$

Thus, under $\boldsymbol{\zeta}$ the product distribution $q_{\mathcal{F}}$ is mapped to the mixture of products $q(\mathbf{x}) = \sum_m q^0(m) q^m(\mathbf{x})$ (after relabelling f^0 to m).

The Lagrangian over the product distribution $q_{\mathcal{F}}(\mathbf{F})$ is $\mathcal{L}$ is

$$
\mathcal{L}(q_{\mathcal{F}}) = \sum_m q^0(m) \mathbb{E}_{q^m}(G) - T\Big[S(q^0) + \sum_{m=1}^{M} S(q^m)\Big].
$$

This Lagrangian offers a term to maximize the entropy of the mixture weights, but it provides no incentive for the distributions q^m to differ from each other. Consequently, we

instead consider the Lagrangian over $q(\mathbf{x})$. In this case

$$\mathcal{L}(q) = \sum_{\mathbf{x}} G\big(\mathbf{x}\big)q(\mathbf{x}) - TS(q) = \sum_{\mathbf{f}} G\big(\boldsymbol{\zeta}(\mathbf{f})\big)q_{\mathcal{F}}(\mathbf{f}) - TS(q)$$
$$= \sum_{m} q^0(m)\mathbb{E}_{q^m}(G) - TS\Big(\sum_m q^0(m)q^m(\mathbf{x})\Big).$$

The entropy term differs crucially in these two variants. To see this more clearly it is convenient to add and subtract $T\sum_m q^0(m)S(q^m)$ to £nd

$$\mathcal{L}(q) = \sum_m q^0(m)\mathcal{L}(q^m) - TJ(q) \tag{8}$$

where $\mathcal{L}(q^m)$ is given by Eq. (1) and where $J(q) \geq 0$ is the Jensen-Shannon (JS) distance,

$$J(q) = S\Big(\sum_m q_0(m)q^m\Big) - \sum_m q_0(m)S(q^m) = -\sum_m\sum_{\mathbf{x}} q_0(m)q^m(\mathbf{x})\ln\frac{q(\mathbf{x})}{q^m(\mathbf{x})}.$$

The JS term is maximized when the q^m are all different from each other and thus pushes the optimal q^m to capture different solutions. Unfortunately, it also couples all variables (because of the sum inside the logarithm), preventing a highly distributed solution. Thus, we replace J with another function which lower-bounds J and which requires less communication between agents.

4.2 A Variational Lagrangian

Following [6], we introduce M variational functions $w(x|m)$ and lower-bound the true JS distance with

$$J(q) = -\sum_m\sum_{\mathbf{x}} q_0(m)q^m(\mathbf{x})\ln\left[\frac{1}{w(\mathbf{x}|m)}q_0(m)\frac{w(\mathbf{x}|m)q(\mathbf{x})}{q_0(m)q^m(\mathbf{x})}\right]$$
$$= \sum_m\sum_{\mathbf{x}} q_0(m)q^m(\mathbf{x})\ln w(\mathbf{x}|m)) - \sum_m q_0(m)\ln q_0(m)$$
$$- \sum_m\sum_{\mathbf{x}} q_0(m)q^m(\mathbf{x})\ln\frac{w(\mathbf{x}|m)q(\mathbf{x})}{q_0(m)q^m(\mathbf{x})}.$$

Now replace M of the $-\ln$ terms with the lower bound $-\ln x \geq -\nu x + \ln\nu + 1$ obtained from the Legendre dual of the logarithm to £nd

$$J(q) \geq J(q,w,\nu) \equiv \sum_m\sum_{\mathbf{x}} q_0(m)q^m(x)\ln w(\mathbf{x}|m) - \sum_m q_0(m)\ln q_0(m)$$
$$- \sum_m \nu_m\sum_{\mathbf{x}} w(\mathbf{x}|m)q(x) + \sum_m q_0(m)\ln\nu_m + 1.$$

Optimization over w and ν maximizes this lower bound. To further aid in distributing the algorithm we restrict the class of variational $w(\mathbf{x}|m)$ to products: $w(\mathbf{x}|m) = \prod_i w_i(x_i|m)$. For this choice

$$J(q,w,\nu) \equiv \sum_m q_0(m)\left\{B^{m,m} - \sum_{\tilde{m}} A^{m,\tilde{m}}\nu_{\tilde{m}} + \ln\nu_m\right\} + S(q_0) + 1 \tag{9}$$

where $A_i^{\tilde{m},m} \equiv \sum_{x_i} q_i^{\tilde{m}}(x_i)w_i(x_i|m)$, $A^{\tilde{m},m} \equiv \prod_{i=1}^d A_i^{\tilde{m},m}$, $B_i^{m,m} \equiv \sum_{x_i} q_i^m(x_i)\ln w_i(x_i|m)$, and $B^{m,m} \equiv \sum_{i=1}^d B_i^{m,m}$.[4] At any temperature T the varia-

[4]Note that if $w_i(x_i|m) = 1/|\mathcal{X}_i|$ is uniform across x_i then $A_i^{\tilde{m},m} = 1/|\mathcal{X}_i|$ and $B_i^{m,m} = -\ln|\mathcal{X}_i|$. Maximizing over ν_m we £nd that $J(q, w = 1/|\mathcal{X}|, \nu = \nu^*) = 0$. Thus, maximizing with respect to w increases the JS distance from 0.

tional Lagrangian which must be minimized with respect to q, w and ν (subject to appropriate positivity and normalization constraints) is then

$$\mathcal{L}(q, w, \nu) = \sum_m q_0(m)\mathcal{L}(q^m) - TJ(q, w, \nu) \tag{10}$$

with $J(q, w, \nu)$ given by Eq. (9).

5 Minimizing the Lagrangian

Equating the gradients with respect to w and ν to zero gives

$$\frac{1}{\nu_m} = \frac{1}{q_0(m)} \sum_{\tilde{m}} q_0(\tilde{m}) A^{\tilde{m},m}. \tag{11}$$

$$w_i(x_i|m) \propto \frac{q_0(m)q_i^m(x_i)}{\nu_m} \left[\sum_{\tilde{m}} q_0(\tilde{m}) q_i^{\tilde{m}}(x_i) \frac{A^{\tilde{m},m}}{A_i^{\tilde{m},m}} \right]^{-1}. \tag{12}$$

The dependence of $\mathcal{L}$ on $q_0(m)$ is particularly simple: $\mathcal{L}(q, w, \nu) \approx \sum_m q_0(m)\mathcal{E}(m) - T\big(S(q_0) + 1\big)$ up to q^0-independent terms and where

$$\mathcal{E}(m) = \mathbb{E}_{q^m}(G) - T\left(S[q^m] + B^{m,m} - \sum_{\tilde{m}} A^{m,\tilde{m}} \nu_{\tilde{m}} + \ln \nu_m \right),$$

Thus, the mixture weights are Boltzmann distributed with energy function $\mathcal{E}(m)$:

$$q_0(m) = \frac{\exp\big(-\mathcal{E}(m)/T\big)}{\sum_{\tilde{m}} \exp\big(-\mathcal{E}(\tilde{m})/T\big)}. \tag{13}$$

The determination of $q_i^m(x_i)$ is similar. The relevant terms in $\mathcal{L}$ involving $q_i^m(x_i)$ are $\mathcal{L} \approx q_0(m) \sum_{x_i} \mathcal{E}_m(x_i) q_i^m(x_i) - TS(q_i^m)$ where

$$\mathcal{E}_m(x_i) = \mathbb{E}_{q_{\setminus i}^m}(G|x_i) - T\left(\ln w_i(x_i|m) - \sum_{\tilde{m}} \frac{A^{m,\tilde{m}}}{A_i^{m,\tilde{m}}} \nu_{\tilde{m}} w_i(x_i|\tilde{m}) \right).$$

As before the conditional expectation $\mathbb{E}_{q_{\setminus i}^m}(G|x_i)$ is $\sum_{x_{\setminus i}} G(x_i, \mathbf{x}_{\setminus i}) q_{\setminus i}^m(\mathbf{x}_{\setminus i})$. The mixture probabilities are thus determined as

$$q_i^m(x_i) = \frac{\exp\big(-\mathcal{E}_m(x_i)/T\big)}{\sum_{x_i} \exp\big(-\mathcal{E}_m(x_i)/T\big)}. \tag{14}$$

5.1 Agent Communication

These results also require minimal communication between agents. An agent, call this the 0-agent, is assigned to manage the determination of $q_0(m)$, and (i, m)-agents manage the determination of $q_i^m(x_i)$. The M (i, m)-agents for a £xed i communicate their $w_i(x_i|m)$ to determine $A_i^{m,\tilde{m}}$. These results along with the $B_i^{m,m}$ from each (i, m) agent are then forwarded to the 0-agent who forms $A^{m,\tilde{m}}$ and $B^{m,m}$ broadcasts this back to all (i, m)-agents. With these quantities and the local estimates for $\mathbb{E}_{q_{\setminus i}^m}(G|x_i)$, all q_i^m can be updated independently.

6 Experiments

We test the probability collective method on two different problems: a k-sat constraint satisfaction problem having multiple feasible solutions, and optimization of an unconstrained optimization of an NK function.

(a) (b)

Figure 1: (a) Evolution of Lagrangian value (solid line), expected constraint violation (dotted line), and constraint violations of most likely con£guration (dashed line). (b) $P(G)$ after minimizing the Lagrangian for the £rst 3 multiplier settings. At termination $P(G) = \delta(G)$.

6.1 *k*-sat

The k-sat problem is perhaps the best studied CSP [7]. The goal is to assign N binary variables x_i values so that C clauses are satis£ed. The ath clause involves k variables labeled by $v_{a,j} \in [1, N]$ (for $j \in [1, k]$), and k binary values associated with each a and labeled by $\sigma_{a,j}$. The ath clause is satis£ed iff $\bigvee_{j=1}^{k}[x_{v_{a,j}} = \sigma_{a,j}]$ is true so we de£ne the ath constraint as

$$c_a(\mathbf{x}) = \begin{cases} 0 & \text{if } \bigvee_{j=1}^{k}[x_{v_{a,j}} = \sigma_{a,j}] \\ 1 & \text{otherwise} \end{cases} .$$

As the ath clause is violated only when all $x_{v_{a,j}} = \overline{\sigma}_{a,j}$ (with $\overline{\sigma} \equiv \text{not}\,\sigma$), the Lagrangian over product distributions can be written as $\mathcal{L}(q) = \boldsymbol{\lambda}^\top \mathbf{c}(q) - TS(q)$ where $c(q)$ is the C-vector of expected constraint violations whose ath component is $c_a(q) \equiv \sum_{\mathbf{x}} c_a(\mathbf{x}) q(\mathbf{x}) = \prod_{j=1}^{k} q_{v_{a,j}}(\overline{\sigma}_{a,j})$, and $\boldsymbol{\lambda}$ is the C vector of Lagrange multipliers. The only communication required to evaluate G and its conditional expectations is between agents appearing in the same clause. Typically, this communication network is sparse; for the $N = 100$, $k = 3$, $C = 430$ variable problem we present each agent interacts with only 6 other agents on average.

We £rst present results for a single product distribution. For any £xed setting of the Lagrange multipliers, the Lagrangian is minimized by iterating Eq. (4). Had the minimization been done by the Brouwer method, any random subset of variables no two of which appear in the same clause could be updated simultaneously while still ensuring that the Lagrangian would decrease at each iteration.

The minimization is terminated at a local minimum q^*. If all constraints are satis£ed at q^* we return the solution $\mathbf{x}^* = \arg\max_{\mathbf{x}} q^*(\mathbf{x})$ otherwise the Lagrange multipliers are updated according to Eq. (7). In the present context, this updating rule offers a number of bene£ts. Firstly, those constraints which are violated most strongly have their penalty increased the most, and consequently, the agents involved in those constraints are most likely to alter their state. Secondly, the Lagrange multipliers contain a history of the constraint violations (since we keep adding to $\boldsymbol{\lambda}$) so that when the agents coordinate on their next move they are unlikely to return a previously violated state. This mimics the approach used

Figure 2: Each constraint's Lagrange multiplier versus the iterations when they change.

in taboo search where revisiting of con£gurations is explicitly prevented, and aids in an ef£cient exploration of the search space. Lastly, rescaling the Lagrangian after each update of the multipliers by $\mathbf{1}^\top \boldsymbol{\lambda} = \sum_a \lambda_a$ gives $\mathcal{L}(q) = \hat{\boldsymbol{\lambda}}^\top \mathbf{c}(q) - \hat{T} S(q)$ where $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}/\mathbf{1}^\top \boldsymbol{\lambda}$ and $\hat{T} = T/\mathbf{1}^\top \boldsymbol{\lambda}$. Since $\sum_a \hat{\lambda}_a$ the £rst term reweights clauses according to their expected violation, while the temperature $\hat{T}$ cools in an automated way as the Lagrange multipliers increase. Cooling is most rapid when the expected constraint violation is large and slows as the optimum is approached. The parameters α_λ^t thus govern the overall rate of cooling. We used the £xed value $\alpha_\lambda^t = 0.5$.

Figure 1 presents results for a 100 variable $k = 3$ problem using a single mixture. The problem is satis£able formula `uf100-01.cnf` from SATLIB (`www.satlib.org`). It was generated with the ratio of clauses to variables being near the phase transition, and consequently has few solutions. 1(a) shows the variation of the Lagrangian, the expected number of constraint violations, and the number of constraints violated in the most probable state $x_{\text{mp}} \equiv \arg\max_{\mathbf{x}} q(\mathbf{x})$ as a function of the number of iterations. The starting state is the maximum entropy con£guration, and the starting temperature is $T = 0.0015$. The iterations at which the Lagrange multipliers are updated are indicated by vertical dashed lines which are clearly visible as discontinuities in the Lagrangian values. To show the stochastic underpinnings of the algorithm we plot in 1(b) the probability density of the number of constraint violations obtained as $P(G) = \sum_{\mathbf{x}} q(\mathbf{x}) \delta\big(G - G(\mathbf{x}, \mathbf{1})\big)$.[5] Figure 2 shows the evolution of the renormalized Langrange multipliers $\hat{\boldsymbol{\lambda}}$. At the £rst iteration the multiplier for all clauses are equal. As the algorithm progresses weight is shifted amongst dif£cult to satisfy clauses.

Results on a larger problem with multiple mixtures are shown in 6.1(a). This is the 250 variable/1065 clause problem `uf250-01.cnf` from SATLIB with the £rst 50 clauses removed so that the problem has multiple solutions. The optimization was performed by at each iteration selecting a random subset of variables, no two of which appear in the same clause and iterating Equations (11), (12), (13), and (14). After convergence the Lagrange

[5] In determining the density 10^4 samples were drawn from $q(\mathbf{x})$ with Gaussians centered at each value of $G(\mathbf{x}, \mathbf{1})$ and with the width of all Gaussians determined by cross validation of the log likelihood. The fact that there is non-zero probability of obtaining non-integral numbers of constraint violations is an artifact of the £nite width of the Gaussians.

(a) (b)

Figure 3: (a) The solid colored curves show the number of unsatis£ed clauses in the most probable con£guration x_{mp} of each of the 4 mixtures vs iterations. The solid black line plots the expected number of violations, and the dashed black line shows the approximation to the JS distance. (b) The solid colored curves show the evolution of the G value of the best x_{mp} con£gurations for each of 5 mixtures versus number of iterations. The dashed black line shows the corresponding approximation to the JS distance.

multipliers are updated. The initial temperature is 0.1. We plot the number of constraints violated in the most probable state of each mixture as a function of the number of updates. as well as the expected number of violated constraints. After 8000 steps three distinct solutions have been found along with a fourth con£guration which violates a single constraint.

6.2 Minimization of NK Functions

The NK model de£nes a family of tunably dif£cult optimization problems [8]. The energy of N binary variables is de£ned as the average of N contributions local to each variable x_i and involving $0 \geq K \geq N-1$ other randomly chosen variables $x_i^1 \cdots x_i^K$: $G(\mathbf{x}) = N^{-1} \sum_{i=1}^{N} E_i(x_i; x_i^1, \cdots x_i^K)$. For each of the 2^{K+1} local con£gurations E_i is assigned a value drawn uniformly from 0 to 1. K controls the number of local minima; under Hamming neighborhoods $K = 0$ optimization landscapes have a single global optimum and $K = N-1$ landscapes have on average $2^N/(N+1)$ local minima. Further properties of NK landscapes may be found in [**?**]. Fig. 6.1(b) plots the energy of a 5 mixture model for a multi-modal $N = 300$ $K = 2$ function. The $K-1$ spins other than i upon which E_i depends were selected at random. At termination of the PC algorithm 5 distinct con£gurations are obtained with the nearest pair of solutions having Hamming distance 12.

7 Conclusion

A distributed constrained optimization framework based on probability collectives has been presented. Motivation for the framework was drawn from an extension of full-rationality game theory to bounded rational agents. An algorithm that is capable of obtaining one or more solutions simultaneously was developed and demonstrated on two problems. The results show a promising, highly distributed, off-the-shelf approach to constrained optimization.

There are many avenues for future exploration. Alternatives to the Lagrange multiplier

method used here can be developed for constraint satisfaction problems. By viewing the constraints as separate objectives, a Pareto-like optimization procedure may be developed whereby a gradient direction is chosen which is constrained so that no constraints are worsened. This idea is motivated by the highly successful WalkSAT [**?**] algorithm for k-sat in which spins are ¤ipped only if no previously satis£ed clause becomes unsatis£ed as a result of the change.

Probability collectives also offer promise in devising new methods for escaping local minima. Unlike traditional optimization methods where monotonic transformations of the objective leave local minima unchanged, such transformations will alter the local minima structure of the Lagrangian. This observation, and alternative Lagrangians (see [**?**] for a related approach using a different minimization criterion) offer new approaches for improved optimization.

References

[1] D. H. Wolpert. Factoring a canonical ensemble into a product of ensembles. 2003. preprint cond-mat/0307630.

[2] D. H. Wolpert. Bounded rational games, information theory, and statistical physics. In D. Braha and Y. Bar-Yam, editors, *Complex Engineering Systems*, 2004.

[3] D. H. Wolpert. Generalizing mean £eld theory for distributed optimization and control. 2004. submitted.

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

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

[6] T. S. Jaakkola and M. I. Jordan. Improving the mean £eld approximation via the use of mixture distributions. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer Academic, 1998.

[7] M. Mezard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satis£ability problems. *Science*, June 2002.

[8] S. A. Kauffman and S. A. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.