

A New Analysis Method for Dynamic, Distributed Constraint Satisfaction

Roger Mailler
University of Tulsa
Tandy School of Computer Science
Tulsa, Oklahoma
mailler@utulsa.edu

Huimin Zheng
University of Tulsa
Tandy School of Computer Science
Tulsa, Oklahoma
huimin-zheng@utulsa.edu

ABSTRACT

There has been an increasing recognition that a number of key computational problems require distributed solution techniques. To facilitate the creation and advancement of these techniques, researchers have developed the distributed constraint satisfaction (DCSP) formalism with the understanding that many critical real-world problems can be represented using it. Consequently, this formalism has led to the creation of myriad protocols for solving problems in this class. However, this formalism ignores a critical feature of many environments: problems change over time.

The dynamic, DCSP (DynDCSP) formalism was invented to address this deficiency, but this model has received inadequate attention from the research community. A key impediment to advancing this research area is the lack of a compelling theoretical underpinning to the analysis of these problems and the evaluation of the protocols used to solve them. This work creates a mapping of the DynDCSP formalism onto thermodynamic systems. Under this mapping, it shows that DynDCSPs obey the three laws of thermodynamics. Utilizing these laws, this work develops, for the first time, a method for characterizing the impact that dynamics has on a distributed problem as well as a technique for predicting the expected performance of distributed protocols under various levels of dynamics.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Performance, Experimentation, Theory

Keywords

Dynamics, Constraint Satisfaction, Thermodynamics

1. INTRODUCTION

Our world is in constant flux. Yet despite this universally accepted truth, most algorithms and protocols are designed with the assumption that problems do not change during

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*
Copyright © 2014, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

the computation of a solution. In many circumstances this assumption is entirely justifiable because either the problem never changes or changes at such a slow rate that a solution can be recomputed each time an alteration occurs without significant degradation of overall performance. However, as the size and difficulty of the real-world problems we design systems to manage has increased, this assumption has become more difficult to justify. Nowhere is this more true than for rapidly-changing, naturally distributed problems where protocol speed is frequently outpaced by the rate that the problem is being altered.

This important class of problems appears in countless multi-agent and multi-robotics papers, however, approaches to addressing it have been mostly adhoc. This is partly caused by the lack of a standardized formulation to develop and evaluate techniques against. Building upon the readily accepted distributed constraint satisfaction problem (DCSP) and with the goal of explicitly modeling time in the formulation, Mailler introduced a model of the dynamic, distributed constraint satisfaction problem (DynDCSP) [8]. Although this work provided a standard formulation to work from, it was only just a beginning because it did little to expand our understanding of protocol construction or even to explain the underlying causes for the performance of the protocols that were presented.

This work makes substantial progress toward understanding the properties of dynamic constraint systems and our ability to analyze and construct protocols that are designed to maintain solution quality as these systems change over time. First, we model DynDCSPs as closed thermodynamic systems where solution quality is equivalent to the energy of the thermodynamic system. By showing that the environment is acting in a manner analogous to applying heat to the system, we then show that the second law of thermodynamics can be used to reliably predict the convergence point and the rate of convergence for systems operating in the absence of a problem solver. We then show that DynCSPs obey the third law of thermodynamics and as a result have solution sets that rapidly decrease in size as the system moves away from the convergence point. Then, by modeling the action of a protocol as work, we present a method for profiling a protocol in the absence of environmental dynamics to measure the work per unit time it performs. Using the first law of thermodynamics, heat and work are then combined together and used to compute the quality equilibrium point. Finally, we show that the equilibrium point is statistically stable.

In the rest of this paper, we will present the formal model

of the dynamic distributed constraint satisfaction problem along with some background in the area of DynDCSPs. In Section 3 we give some background on the connection between statistical physics and constraint satisfaction and present a mapping of DynCSPs to thermodynamic systems. Section 4 shows that DynCSPs obey the second law of thermodynamics, which can be used to predict their most likely state in the presence of environmental dynamics. Section 5 shows that the third law of thermodynamics also applies to DynDCSPs. This law helps us to understand how the solution state changes as we move further from the convergence point. In Section 6, we show that the first law of thermodynamics can be used to predict the performance of a distributed protocol operating in a dynamic environment. Finally, in Section 7 we present our conclusions and future directions.

2. BACKGROUND

Formally speaking, a static DCSP, $P = \langle V, A, D, C \rangle$, consists of the following [25]:

- a set of n variables: $V = \{v_1, \dots, v_n\}$.
- A set of g agents: $A = \{a_1, \dots, a_g\}$.
- Discrete, finite domains for each variable:
 $D = \{D_1, \dots, D_n\}$.
- A set of constraints $C = \{c_1, \dots, c_m\}$ where each $c_i(d_{i,1}, \dots, d_{i,j})$ is a function $c_i : D_{i,1} \times \dots \times D_{i,j} \rightarrow \{true, false\}$. i.e. the constraints take in values for the variables and return true if the constraint is satisfied.

The task is to find an assignment $S^* = \{d_1, \dots, d_n | d_i \in D_i\}$ that is free of constraint violations or report that no such solution exists. Each agent in the set A is assigned one or more variables along with constraints associated with its variables. The goal of each agent, from a local perspective, is to ensure that it gets a violation free solution for its variables.

Based on this definition, we can see that DCSPs are parameterized based on the values of n , m , g , and j . This paper frequently assumes that the constraints are all binary and therefore all have $j = 2$. However, this restriction does not limit the usefulness of the results because n -ary constraints can be easily converted to binary constraints [1]. In addition, DCSP instances are frequently classified based on their density and tightness. The density, p_1 , of a DCSP is the ratio of the number of constraints to possible number of constraints (i.e. $p_1 = \frac{2^*m}{n*(n-1)}$). The tightness, p_2 , is the ratio of the number of assignments in a constraint that return *false* to the total number of assignments.

Since its creation, the DCSP formalisms have been at the center of research in distributed problem solving. As such, it has spawned the creation of protocols including the distributed breakout algorithm (DBA), [24], asynchronous backtracking (ABT) [23], the distributed stochastic algorithm (DSA) [27], and asynchronous partial overlay (APO) [10].

This formalism and all of the algorithms that have been created for solving it share one common property: they describe and are intended to solve static problems. For some domains, this limitation does not significantly detract from the usefulness of the DCSP definition. However many practical uses for distributed protocols involve environments that

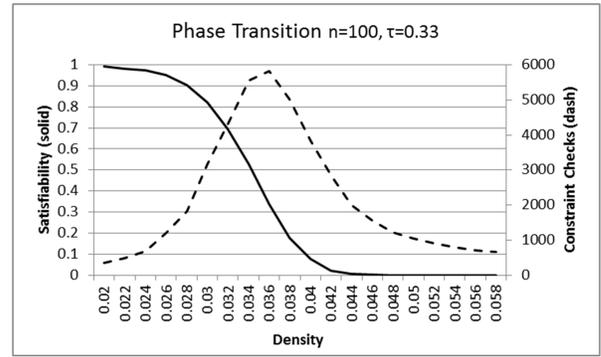


Figure 1: Phase transition results for $n = 100$ and $p_2 = 0.33$.

change over time. These include distributed resource allocation [2], distributed scheduling [20], and distributed planning [19].

Strangely, research on dynamic DCSPs is fairly sparse. Formally, a dynamic DCSP (DynDCSP) is a sequence of DCSPs $\{P_0, P_1, \dots, P_n\}$ where each P_i is a static DCSP as defined above [3]. If we define c_i^a as a set of added constraints and c_i^r as a set of removed constraints functions, then $P_i = P_{i-1} + c_i^a - c_i^r$ [21].

The centralized techniques that have been developed for solving these problems fall into two categories: reactive and proactive [21]. Reactive methods make no assumptions about how a problem is going to change from one instance to the next. Instead they use either local modifications or prior knowledge to repair an existing solution. Proactive methods, on the other hand, attempt to build robust or flexible solutions that have a high probability of either remaining consistent or being easily altered if the problem changes. An excellent review of this field of research can be found in the work of Verfaillie and Jussien [21].

The protocols that have been designed for DynDCSPs have all been reactive. For example, in the work of Modi et al. [25], the authors present a novel algorithm called Locally Dynamic AWC (LD-AWC) that is both complete and sound given that only non-shared (unary) constraints change. Fitzpatrick and Meertens present an incomplete, hill-climbing method for solving dynamically changing graph coloring problems [5] that strongly resembles the distributed stochastic algorithm (DSA) [27]. More recently dynamic versions of the DBA, AWC, and APO protocols have been created [8, 11].

Although the DynDCSP formalism is one step closer to describing computational systems that operate "in the wild", they fail to address the relationship between changes to the problem and real-time. Specifically, these formulations ignore the rate of change (as related to real-time) between instances, instead assuming that there is sufficient time between each change to compute a new solution.

To rectify this deficiency, Mailler modified the basic definition of a DynDCSP by introducing a function $\Delta : P_t \mapsto P_{t+1}$ that maps a DCSP at time t to a new DCSP at $t + 1$ by adding and removing constraints [8]. With the Δ function in place, we can then begin to measure the amount of change a problem experiences over unit time by counting the number of constraints that are added and removed. In other

words, the rate of change of a problem is defined as

$$\begin{aligned} \text{rate} &= \frac{dP}{dt} = \lim_{\Delta t \rightarrow 0} \frac{P_{t+\Delta t} - P_t}{\Delta t} \\ &= \frac{1}{\Delta t} \sum_{i=t}^{t+\Delta t} (|c_i^a| + |c_i^r|) \end{aligned} \quad (1)$$

By relating the change of a problem to time in this manner, this work recognized that a problem may change before an algorithm is able to generate a full solution. It also means that for DynDCSPs, instantaneous quality becomes related to the number of constraints that are satisfied making them more akin to a maximum satisfiability problem. Because of this, one should realize that as $\frac{dP}{dt}$ increases, more reactive and potentially less optimal problem solvers may actually be better.

3. DynDCSPs AS THERMODYNAMIC SYSTEMS

There has been an increasing amount of evidence that supports a strong analogy between NP-complete problems and disordered systems in physics. The earliest recognition that this connection may exist can be found in the work of Kirkpatrick, Gelatt, and Vecchi [7]. By modeling a traveling salesman problem (TSP) as a set of atoms at equilibrium, they were able to apply what was then known as the Metropolis algorithm [12] to modify the temperature to arrive at good solutions. We now refer to this technique as simulated annealing.

Since then, the connection between computational problems and statistical physics has only gotten stronger. For example, Mezard, and Parisi used advances in spin-glass theory to provide analytical results for the weighted matching problem [13] and later for TSP [14]. They were able to show that randomly generated problems undergo a phase transition as the ratio of nodes to edges is varied. Following up on this work, Monasson and Zecchina were able to apply the same basic modeling techniques to relate an order parameter $\alpha = \frac{m}{n}$ to the number of satisfying solutions for the well-known NP complete problem, K-satisfiability [15, 16]. Their pioneering work led to the recognition that problems within the phase transition exhibit backbone variables. These variables are critically important because they only have one allowable value in all possible solutions to the problem. From the computational perspective this means that the difficulty of computing a solution to an NP-complete problem is strongly associated with the phase transition [17].

Figure 1 shows the results of a phase transition analysis conducted on random CSP problems with $n = 100$ variables, a domain size of 3, and a tightness of $p_2 = 0.33$. Each data point represents an average over 10,000 separate graph instances. As can be seen in this graph, the problems rapidly go from being satisfiable to unsatisfiable over a very small density range. At the same time, the number of constraint checks performed by a Forward Checking-Conflict Directing Backjumping algorithm [18] reaches its peak at the %50 satisfiable/unsatisfiable point. Throughout this work we use graphs with these settings and have chosen three density values that create satisfiable ($p_1 = 0.02$), phase-transition ($p_1 = 0.035$), and unsatisfiable ($p_1 = 0.055$) instances.

With such a strong analogy between static CSPs and statistical physics, the connection between DynDCSPs and ther-

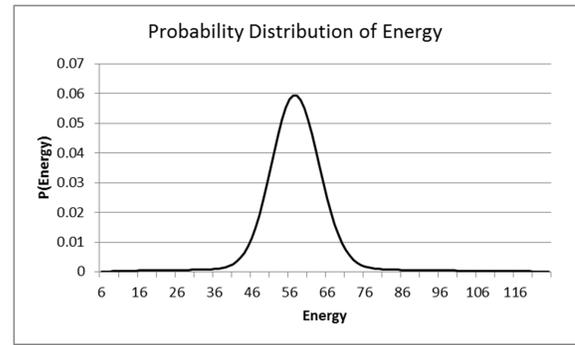


Figure 2: Energy distribution for $n = 100$, $p_1 = 0.035$, and $p_2 = 0.33$

modynamics certainly seems plausible. Following the approach of Zdeborova and Krzakala [26], we can map a CSP or DCSP into a spin model where each variable v_i maps to an equivalent spin, s_i . We can say that, based on the domain of v_i , that s_i can only be in one of $q = |D_i|$ different quantum states.

Using this as a basis, if we consider only binary constraints that have a cost of 1 when unsatisfied then $c_k(d_i, d_j) \mapsto \{0(\text{true}), 1(\text{false})\}$ then the Hamiltonian becomes

$$\mathcal{H}(\mathcal{S}) = \sum_{c_k \in \mathcal{C}} c_k(d_i, d_j) = E \quad (2)$$

In statistical physics, the Hamiltonian is a measure of the energy of the system and in our mapping measures the number of constraints that are violated by having the variables in the state \mathcal{S} . Under this definition the energy can become zero if and only if the DCSP is satisfiable. In many cases, a large number of states will have a given energy E . We can define a unit-less value $\Omega(E)$, which is often simply denoted as Ω , to be the total number of states with a given energy level and further define $S = k \ln \Omega$ as the entropy of the system. The constant k rescales and gives a unit to entropy. Information theory uses bits as its unit, while in physics the standard unit is joules/kelvin. The particular unit of entropy is not important for our purposes.

To extend this mapping to show that DynDCSPs can be modeled as thermodynamic systems, one must recognize that environmental dynamics, by adding and removing constraints, is equivalent to adding and removing heat from the problem. These changes may subsequently result in a shift to the energy (solution quality) and entropy (number of solutions with a given energy) of the system. In the limited case where the heat being added and removed are equal, the entropy of the system remains stationary although the energy level may still change. In fact, as long as a DynDCSP is only being modified by altering the constraints we can show that the energy of a DynDCSP always converges to a predictable value.

4. SECOND LAW OF THERMODYNAMICS

The proof of the convergence is based on Boltzmann's famous H-theorem [4]. Let us say that at any given time that a DynDCSP can be in one of many states. We can label one of these states as r and say the prior probability of finding the system in state r at any given time is P_r . The

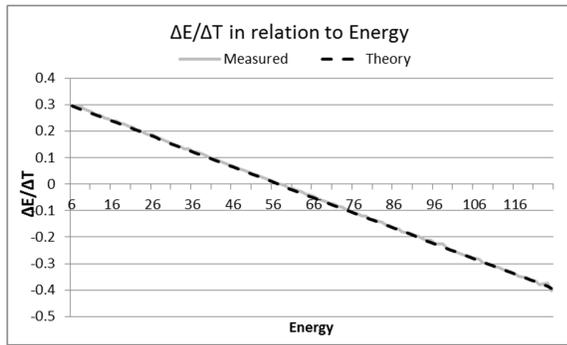


Figure 3: Change rate of energy relative to current energy

value of P_r is affected by two things: either a system in state r can transition to state s or a system in state s , with prior probability P_s can transition to state r . The H-theorem shows that as long as the probability of transitioning from state r to state s is equal to the probability of transitioning from state s to state r then the system will statistically converge onto the state with the highest prior probability. In other words, as long as the probabilities of adding and removing an individual constraint are the same, then the system will converge onto its most likely state.

Boltzmann did this proof by introducing a quantity $H \equiv \ln P_r = \sum_r P_r \ln P_r$ and showed that $\frac{dH}{dt} \leq 0$. Later, Gibbs defined $S = -K_B \sum_r P_r \ln P_r$ where K_B is a constant. This means that $S \approx -H$ and, therefore if the H-theorem holds for a system, which it does in our case, then $\frac{dS}{dt} \geq 0$ is also true. This is the second law of thermodynamics which states:

Definition 1. Second Law of Thermodynamics - The entropy of an isolated system can never spontaneously decrease [4].

For binary DynDCSPs, the expectation of the energy E , which represents the most likely state of the system, can be calculated using the linearity of expectation and the definition of tightness:

$$\begin{aligned}
 \langle E \rangle &= \left\langle \sum_{c_k \in C} c_k(d_i, d_j) \right\rangle \\
 &= \sum_{c_k \in C} \langle c_k(d_i, d_j) \rangle \\
 &= \sum_{c_k \in C} p_2 = mp_2
 \end{aligned} \tag{3}$$

Intuitively, this result makes sense because the energy of the system is strictly associated with the number of constraints and the prior probability that an individual constraint is violated, which is the definition of p_2 . To demonstrate that the H-theorem empirically holds and that the calculation for the expectation of energy is correct, we conducted 10,000 experiments each of 5,000 steps where we generated a random DCSP with $n = 100$ variables, $p_1 = 0.035$, and $p_2 = 0.33$. After generating the problem, we computed two assignments for the variables used a simple hill-climbing

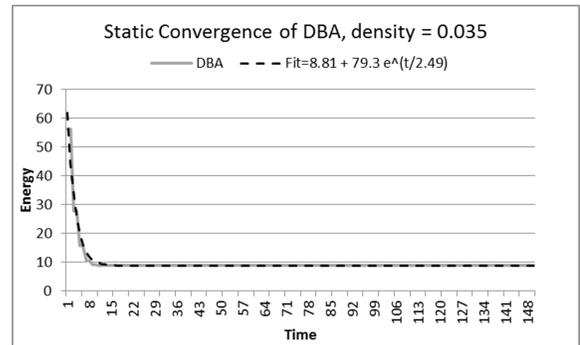


Figure 4: Converge profile of DBA for $n = 100$, $p_1 = 0.035$, and $p_2 = 0.33$

algorithm. The first assignment attempted to put the problem in a ground state, i.e. solve the problem. The other assignment attempted to generate an anti-solution by maximizing the energy level. With these assignments in hand, at each time step of the simulation, we perturbed the problem by removing an existing constraint and replacing it with a new one. We then calculated the new energy of both assignments. We recorded the number of occurrences at each energy level as the simulation progressed.

Figure 2 shows the results of this experiment. For a problem with 100 variable and density of 0.035, there will be 173 constraints. With a $p_2 = 0.33$ we would expect the system to statistically converge to a mean energy of $mp_2 = 173 * 0.33 = 57$. These results clearly show that the distribution has a most likely value of 57 as the theory predicts.

Knowing the system convergence point is important, but it is probably more important to be able to calculate the rate of convergence. In thermodynamics, this is measured as the expected change in energy per unit time or $\frac{\langle \Delta E \rangle}{\Delta t}$. To simplify the analysis, let $\Delta t = 1$, so that during a single time step one constraint is selected and replaced with a new constraint. This implies that the energy can change by at most 1 during each time step. Let's also define P^+ to be the probability that the energy level increases, P^- to be the probability that it decreases, and P^0 to be the probability that it remains unchanged. This gives us

$$\frac{\langle \Delta E \rangle}{\Delta t} = \langle \Delta E \rangle = 1 * P^+ + 0 * P^0 + (-1) * P^- \tag{4}$$

Since it is unimportant to consider the value of P^0 because it does not impact the expected change in energy, let us consider P^+ and P^- . P^+ is the probability of selecting a currently satisfied constraint and replacing it with one that is not satisfied. Well, the probability of selecting a satisfied constraint is just the ratio of current satisfied constraints to total number of constraints. Also, the probability of randomly generating an unsatisfied constraints is given by p_2 therefore:

$$P^+ = \frac{m - \langle E \rangle}{m} * p_2 \tag{5}$$

Similarly P^- is the probability of selecting a currently violated constraint and replacing it with one that is satisfied:

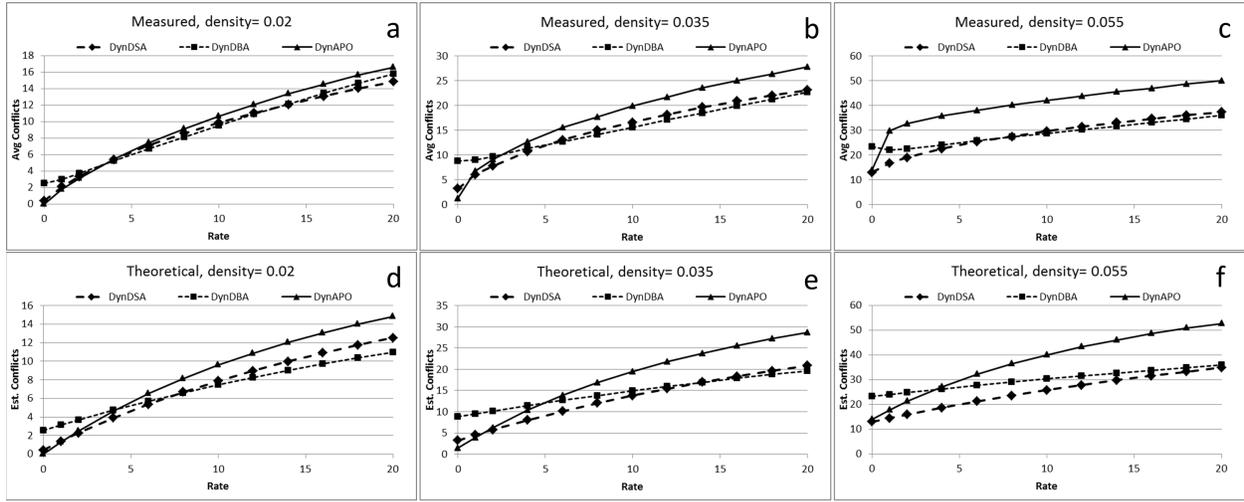


Figure 5: Measured and theoretical equilibrium of DynDSA, DynDBA, and DynAPO

$$P^- = \frac{\langle E \rangle}{m} * (1 - p_2) \quad (6)$$

By substituting Equations 5 and 6 into Equation 4 and converting it to a differential equation we get

$$\frac{dE}{dt} = \frac{m - E}{m} p_2 - \frac{E}{m} (1 - p_2) = p_2 - \frac{E}{m} \quad (7)$$

This differential equation has a closed form solution of

$$E = (E_0 - mp_2)e^{-\frac{t}{m}} + mp_2 \quad (8)$$

This formula has an identical structure to Newton's cooling law and can be used to predict the change in the energy level of a DynDCSP over time or, in essence, to quantify the expected impact that the environment has on a problem.

To empirically validate our theoretical results, we conducted the same experiment as described above, but this time measured the change in energy per unit time in relation to the energy level. Figure 3 shows the results of this experiment. The grey line shows the measured average change in energy, while the dotted black line shows $\frac{dE}{dt}$ as predicted by the equation above. The match is a perfect fit. Note that the line crosses the origin at precisely $mp_2 = 57$ and that for values of energy less than 57, the energy is expected to increase, while the opposite is true for values greater than 57. From these results it appears that our model perfectly predicts the expected behavior of the DynDCSP.

5. THIRD LAW OF THERMODYNAMICS

The H-theorem and its relationship to entropy tell us quite a bit about the solution space for a DynDCSP. When a DynDCSP is not in its most likely state, with an energy of mp_2 , we know from the H-theorem that $\frac{dH}{dt} \leq 0$ and given the relationship between $\frac{dH}{dt}$ and $\frac{dS}{dt}$ we can further assume that $\frac{dS}{dt} \geq 0$. This tells us that as we move further from the most likely state that S is monotonically decreasing and therefore must reach it lowest value as it approaches an energy of 0 or m . However, it tells us little about how fast the solution space decreases. This is provided by the Third Law of Thermodynamics.

Definition 2. Third Law of Thermodynamics - In the limit as the absolute temperature tends to zero the entropy also tends to zero [4].

To show the Third Law holds, we must first provide an estimate for the function Ω . Based on the principle of equal *a priori* probabilities, we can say that $P(E) = C\Omega$, where C is a constant that is not associated with the energy of the system. For DynDCSPs the probability of exactly E constraints being violated is given by the binomial distribution:

$$P(E) = \binom{m}{E} p_2^E (1 - p_2)^{C-E} \propto \Omega \quad (9)$$

This function is the number of ways to choose E constraints out of m , times the probability of having exactly E out of m constraints violated. We will use the classical definition for thermodynamic temperature, which is

$$\frac{1}{kT} = \beta = \frac{\partial \ln \Omega}{\partial E} \quad (10)$$

In this equation, it is important to note that temperature T is similar to S , as it is unit-less, based on the number of potential states of the system, and is given units using the constant k . We introduced the variable β as a convenience as it should be understood that $\frac{1}{T} \propto \beta$. Since we know that the energy level in DynDCSPs take on integer values, we can estimate the value of β by measuring $\beta(E) \approx \ln \Omega(E) - \ln \Omega(E-1)$. Substitution of our estimate of Ω gives us:

$$\begin{aligned} \beta(E) &\approx \ln \left(\binom{m}{E} p_2^E (1 - p_2)^{C-E} \right) - \ln \left(\binom{m}{E-1} p_2^{E-1} (1 - p_2)^{C-E+1} \right) \\ &= \ln \frac{\binom{m}{E} p_2^E (1 - p_2)^{C-E}}{\binom{m}{E-1} p_2^{E-1} (1 - p_2)^{C-E+1}} \\ &= \ln \left(\frac{m!}{E!(m-E)!} \cdot \frac{(E-1)!(m-E+1)!}{m!} \cdot \frac{p_2}{1-p_2} \right) \\ &= \ln \left(\frac{C+1}{E} - 1 \right) + \ln \left(\frac{p_2}{1-p_2} \right) \end{aligned} \quad (11)$$

So, as $T \rightarrow 0$, we would expect that $\beta(E) \rightarrow \infty$ while $E \rightarrow 0$. We also know from the second law that as $E \rightarrow 0$, which is the furthest value for E less than mp_2 , that $S \rightarrow 0$. Therefore as $T \rightarrow 0$, $S \rightarrow 0$. So, the Third Law holds under our mapping.

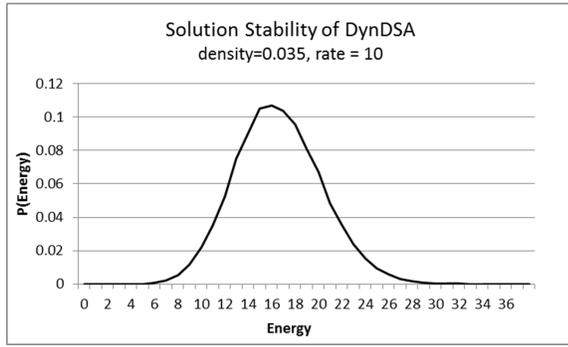


Figure 6: Equilibrium stability for DynDSA with $n = 100$, $p_1 = 0.035$, $p_2 = 0.33$, and $rate = 10$

6. FIRST LAW OF THERMODYNAMICS

Probably the most useful law for characterizing an environment and being able to predict the performance of a protocol is the First Law of Thermodynamics which states:

Definition 3. First Law of Thermodynamics - The change in internal energy of a system is the difference between the net heat absorbed by the system from its surroundings and the net work done by the system on its surroundings [4].

This law can be directly translated into the famous equation $\Delta E = \delta Q - \delta W$ where δQ is the heat being applied to the system and δW is the work done by or on the system. For our purposes, this law is important because it can be used to predict the energy equilibrium point by determining the value for E where $\Delta E = 0$ or in other words when $\delta Q = \delta W$.

We have already shown that using our mapping we can calculate the value of δQ , which is measured as $\frac{dE}{dt}$, and that, in the absence of a solver, $\frac{dE}{dt}$ converges to 0 as $E \rightarrow mp_2$. Recall from Section 4 that

$$\delta Q = \frac{dE_Q}{dt} = p_2 - \frac{E}{m} = \frac{mp_2 - E}{m} \quad (12)$$

What remains to be shown is that we can calculate δW and use it to predict the change in energy, show that the system converges to an equilibrium, and determine that equilibrium point. Let us assume that the First Law applies to DynDCSPs. This means that, if a DynDCSP remains static with $Q = 0$, that the energy profile created while solving this static instance is a direct measure of the work being done by a solver. So, by running a solver on static DCSP instances and measuring the rate of change in the energy, we are in fact measuring W , and can create a function that will predict the energy of the system at time t . Let's suppose that this function looks similar to the cooling function and takes the form

$$E_W = B + Ne^{-\frac{t}{A}}, 0 < B < mp_2, 0 < N \quad (13)$$

However, we are interested in the change in energy for the current energy, which is given by a function with a familiar form:

$$\frac{dE_W}{dt} = \frac{B - E}{A} \quad (14)$$

The idea of fitting the convergence profile to an exponential function is certainly reasonable if one considers that protocols work in parallel. This implies that the number of corrected constraints per unit time is expected to be directly proportional to the number of current violations. In fact, to validate our assumption, we ran a series of tests that measured the average (sample size of 50 instances) change in energy over time of three, state-of-the-art, distributed protocols using randomly generated 100 variable, static instances with $p_1 = \{0.02, 0.035, 0.055\}$ and $p_2 = 0.33$. For the study, we used adapted versions of DSA [5, 27], DBA [24], and APO [10, 9]. Again, these density values were chosen because they are expected to lie within the three regions of the phase transition for problems of size $n = 100$.

Figure 4 shows the average energy profile for DBA operating on problems of $p_1 = 0.035$. Using Mathematica [22], we produced an extremely close fit using the function $E_W = 8.81 + 79.3e^{-\frac{t}{2.49}}$. In a similar manner, all of the protocols were fit with high precision using the form of the cooling rate function.

By combining Equations 12 and 14 into the First Law, we can calculate the equilibrium point using the equation

$$\frac{mp_2 - E}{m} + \frac{B - E}{A} = 0 \quad (15)$$

which, when solved for E gives us

$$E_0 = \frac{Amp_2 + mB}{A + m} \quad (16)$$

This equation allows us to predict the equilibrium point under the assumption that only one constraint changes per unit time. If, as the definition for the DynDCSP allows, more than one constraint changes at a time or there are several time steps between changes, the cooling/heating equations need to be altered as follows

$$E = mp_2 - be^{-\frac{rate * t}{m}} \quad (17)$$

This then leads to the more general energy equilibrium equation for DynDCSPs which becomes

$$E_0 = \frac{Amp_2 + \frac{mM}{rate}}{A + \frac{m}{rate}} \quad (18)$$

To show that these findings match empirical results, we again ran a simulation with $n = 100$ variables, $p_2 = 0.33$, and densities= $\{0.02, 0.035, 0.055\}$. This time we varied the problems over time by adding and removing an equal number of constraints during each time step according to a rate. The problems were run for 1,000 time steps and the average energy measured. Each data point in the measured graphs represents the average of 50,000 separate measurements. The graphs in Figure 5 show the results of this experiment along with the theoretically predicted values based on Equation 18.

Density	0.02	0.035	0.055
DynDSA	1.81	2.43	3.15
DynDBA	2.60	1.42	1.57
DynAPO	1.15	1.52	5.92

Table 1: RMS Error for the predictions of the energy equilibrium

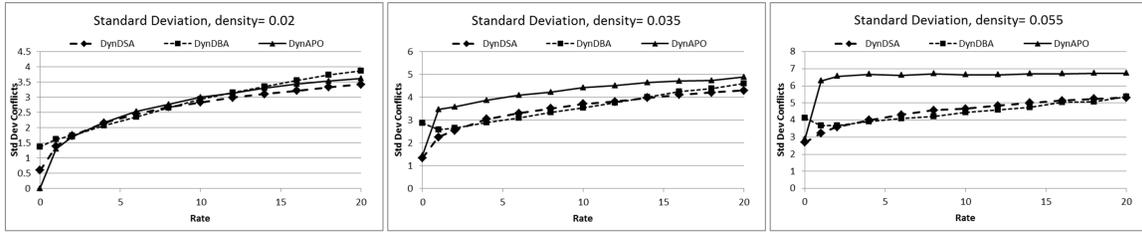


Figure 7: Standard Deviation of Equilibrium for DynDSA, DynDBA, and DynAPO with $n = 100$, $p_2 = 0.33$

The theoretical and empirical results are a very close match as a pairwise comparison of the graphs in Figure 5 reveal. To show how well the predictions work, we computed the RMS error for each protocol at the various densities. As can be seen in Table 1, the error is very low with the highest value occurring for DynAPO on high density, unsatisfiable problems. This prediction error occurs because DynAPO is designed to “give up” when it encounters an unsatisfiable subproblem and accept a local variable value that minimizes the number of constraint violations. Although this strategy works in practice, it makes it harder to predict the equilibrium point of the protocol based on static instances.

6.1 Equilibrium Stability

Although our mapping allows the average energy to be computed when both the environment is changing and a protocol is operating to solve a problem, it is also important to understand if the system operates in a stable manner. To show that the equilibrium energy is in fact a stable value, we can use the the contraction mapping theorem [6]. Let the function $G(E) = E + \frac{mp_2 - E}{m} + \frac{B - E}{A}$. Using Equation 15, we know that $G(E_0) = E_0$. Now assume some small deviation ΔE occurs that moves the system to a new energy level. We can compute the change in $G(E)$ as $\Delta G(E)$, which will be

$$\Delta G(E_0 + \Delta E) = |G(E_0 + \Delta E) - G(E_0)| \quad (19)$$

By substitution and replacing $G(E_0) = E_0$ the equation becomes

$$\begin{aligned} & \Delta G(E_0 + \Delta E) \\ &= \left| G(E_0 + \Delta E) + \frac{mp_2 - (E_0 + \Delta E)}{m} + \frac{B - (E_0 + \Delta E)}{A} \right| \\ &= \left| \Delta E - \frac{E}{m} - \frac{\Delta E}{A} \right| \end{aligned} \quad (20)$$

Since $\Delta G(E_0 + \Delta E)$ will always be less than ΔE as long as $m > 0$, the system must contract back to the equilibrium value E_0 .

To empirically show that the equilibrium is stable, we processed the data from our previous empirical evaluation. This time, we computed a histogram showing the number of occurrences of each energy level while DynDSA was running on problems of size $n = 100$, $p_1 = 0.035$, and $p_2 = 0.33$ that were changing at $rate = 10$. Figure 6 shows the results of this evaluation. From the graph it appears that, like the convergence tests from Section 4, the protocol converges on its most likely state of E_0 . To show that this also occurs for the other protocols Figure 7 shows the standard deviations for all of the test conditions we explored. As can be seen in

this figure, the solution qualities remain quite stable, despite the fact that problems begin to change quite rapidly and, as Figure 5 showed, the solution quality degrades. These results certainly support the theoretical findings quite well.

7. CONCLUSIONS

The dynamic, distributed constraint satisfaction problem occurs in many real-world settings. However, little work has been done on this important problem because the protocols are difficult to develop and analyze. In this paper, we presented a new method for approaching this complex task by mapping the DynDCSP onto physical system and then showing that they obey the three laws of thermodynamics. As a result, we were able to show that you can classify DynDCSPs based on their expected convergence point and rate of convergence. You can also measure the performance of protocols based on the amount of work per unit time they perform. Combining these two results together yields, for the first time ever, a way to predict the performance of a protocol operating in a dynamic environment under previously untested conditions. Finally, this work shows that dynamic protocols form stable equilibria with their environment even as the environment changes very rapidly.

It should be clear that the findings in the work are very powerful and the impact extends outside of the distributed problem solving research discipline. However, additional work must be done to cover other real world situations. For example, the theory should be expanded to cover the addition and removal of domain elements, changes to the density of the problem, or alterations to the tightness of the constraints. In addition, many real-world problems cannot be addressed using binary constraints and therefore must be expanded to deal with cost functions between variables. We are also planning to expand the scope and depth of our empirical evaluations to include additional algorithms, a larger range of problems, and a greater number of samples.

Finally, it is our firm belief that this theory along with the supporting evidence will open the door to discovering new protocols that are tailored to operate in dynamic environments. This is particularly important because, we believe, distributed solutions are likely to produce better overall solution quality than centralized approaches in environments that change rapidly and have substantial communication delay.

8. ACKNOWLEDGMENTS

This material is based on research sponsored by the Air Force Research Laboratory, under agreement number FA8750-13-1-0124. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwith-

standing any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Research Laboratory or the U.S. Government.

9. REFERENCES

- [1] F. Bacchus and P. van Beek. On the conversion between non-binary constraint satisfaction problems. *Proceedings of the 15th Nat'l/10th Conf. on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, pages 311–318, 1998.
- [2] S. E. Conry, K. Kuwabara, V. R. Lesser, and R. A. Meyer. Multistage negotiation for distributed constraint satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6), Nov. 1991.
- [3] R. Dechter and A. Dechter. Belief maintenance in dynamic constraint networks. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-88)*, pages 37–42, 1988.
- [4] R. Fitzpatrick. *Thermodynamics and Statistical Mechanics: An intermediate level course*. Lulu Enterprises, Inc., 2006.
- [5] S. Fitzpatrick and L. Meertens. *Distributed Sensor Networks: A Multiagent Perspective*, chapter Distributed Coordination Through Anarchic Optimization, pages 257–294. Kluwer Academic Publishers, 2003.
- [6] A. Granas and J. Dugundji. *Fixed Point Theory*. Springer-Verlag, New York, 2003.
- [7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [8] R. Mailler. Comparing two approaches to dynamic, distributed constraint satisfaction. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 1049–1056, 2005.
- [9] R. Mailler. Improving asynchronous partial overlay. In *Proceeding of IAT 2012*, 2012.
- [10] R. Mailler and V. Lesser. Asynchronous Partial Overlay: A new algorithm for solving distributed constraint satisfaction problems. *Journal of Artificial Intelligence Research*, 25:529–576, 2006.
- [11] K. Mertens, T. Holvoet, and Y. Berbers. The dyncooa algorithm for dynamic constraint optimization problems. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2006.
- [12] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teher, and E. Teher. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.
- [13] M. Mezard and G. Parisi. Replicas and optimization. *Journal of Physics Letters*, 46, 1985.
- [14] M. Mezard and G. Parisi. A replica analysis of the traveling salesman problem. *Journal of Physics*, 47, 1986.
- [15] R. Monasson and R. Zecchina. Entropy of k-satisfiability problems. *Physical Review Letters*, 76:3881–3885, 1996.
- [16] R. Monasson and R. Zecchina. The statistical mechanics of the random k-satisfiability model. *Physical Review E*, 56:1357–1370, 1997.
- [17] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky. Determining computational complexity from characteristic ‘phase transitions’. *Nature*, 400:133–137, 1999.
- [18] P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 9(3):269–299, 1993.
- [19] O. Sapena, E. Onaindia, A. Garrido, and M. Arangu. A distributed {CSP} approach for collaborative planning systems. *Engineering Applications of Artificial Intelligence*, 21(5):698 – 709, 2008.
- [20] K. Sycara, S. Roth, N. Sadeh, and M. Fox. Distributed constrained heuristic search. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(6):1446–1461, November/December 1991.
- [21] G. Verfaillie and N. Jussien. Constraint solving in uncertain and dynamic environments: A survey. *Constraints*, 10(3):253–281, 2005.
- [22] I. Wolfram Research. Mathematica version 8.0, 2010.
- [23] M. Yokoo, E. H. Durfee, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for formalizing distributed problem solving. In *International Conference on Distributed Computing Systems*, pages 614–621, 1992.
- [24] M. Yokoo and K. Hirayama. Distributed breakout algorithm for solving distributed constraint satisfaction problems. In *Proceedings of the 2nd Int'l Conf. on Multiagent Systems*, pages 401–408, 1996.
- [25] M. Yokoo, T. Ishida, and K. Kuwabara. Distributed constraint satisfaction for dai problems. In M. Huhns, editor, *Proceedings of the 10th Workshop on Distributed Artificial Intelligence*, 1990.
- [26] L. Zdeborova and F. Krzakala. Phase transitions in the coloring of random graphs. *Physical Review E*, 76, 2007.
- [27] W. Zhang, G. Wang, Z. Xing, and L. Wittenburg. Chapter 13: A comparative study of distributed constraint algorithms. *Distributed Sensor Networks: A Multiagent Perspective*, 2003.