INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Information Theory and the Finite-Time Behavior of the Simulated Annealing Algorithm: Experimental Results

Mark Fleischer, Sheldon H. Jacobson,

To cite this article:
Mark Fleischer, Sheldon H. Jacobson, (1999) Information Theory and the Finite-Time Behavior of the Simulated Annealing Algorithm: Experimental Results. INFORMS Journal on Computing 11(1):35-43. http://dx.doi.org/10.1287/ijoc.11.1.35

INFORMS Journal on Computing
Vol. 11, No. 1, Winter 1999

0899-1499/99/1101-0035 $05.00

Information Theory and the Finite-Time Behavior of the Simulated Annealing Algorithm: Experimental Results

MARK FLEISCHER / *Department of Engineering Management, Old Dominion University, Norfolk, VA 23529-0248, Email: mfleisch@odu.edu*

SHELDON H. JACOBSON / *Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061-0118, Email: jacobson@vt.edu*

(Received: January 1994; revised: April 1996, July 1998; accepted September 1998)

This article presents an empirical approach that demonstrates a theoretical connection between (information theoretic) entropy measures and the finite-time performance of the simulated annealing algorithm. The methodology developed leads to several computational approaches for creating problem instances useful in testing and demonstrating the entropy/performance connection: use of generic configuration spaces, polynomial transformations between *NP*-hard problems, and modification of penalty parameters. In particular, the computational results show that higher entropy measures are associated with superior finite-time performance of the simulated annealing algorithm.

In recent years, several articles have appeared describing the problems and issues associated with the propriety of empirical methods, their limitations, and the appropriate manner for demonstrating empirical results.[2, 10, 11, 15, 17, 18, 20, 23] Often, these issues concern comparisons between *different algorithms given a problem instance.* The competitive testing of algorithms[11] naturally arises in these contexts. Although much can be learned from such testing and comparisons, similar issues and benefits can also be derived from comparisons based on the application of *different problem instances to a given algorithm.* This *reverse approach* can be used to discern theoretical relationships between the *performance* of an algorithm and *the characteristics of problem instances* so as to gain an understanding of the inner workings of an algorithm that are not otherwise apparent. In particular, this article presents a methodology, together with computational results that demonstrate a theoretical relationship between entropy measures and the finite-time performance of the simulated annealing (SA) algorithm.

The SA algorithm has been a valuable tool for tackling *NP*-hard combinatorial optimization problems (COPs). SA has been studied extensively to understand its behavior on real-world problems and to identify methods to improve its performance. Johnson et al.[13] observe that it is desirable to have smooth configuration spaces that permit easy escape from local minima. On the other hand, Goldstein and Waterman[9] suggest that too many neighbors may actually hinder SAs performance, as too many neighbors tend to smooth out the configuration space by reducing the number of local optima, thereby allowing easy escape from the global optima. This suggests two conflicting points of view, hence motivates a search for properties of configuration spaces linked to the finite-time performance of SA. Ideally, a single measure on a configuration space that captures the *entire* topology is desired (e.g., all of its smoothness, hilliness; see [21, p. 154] for a measure denoted as "space conductance").

Fleischer and Jacobson[5] and Fleischer[4] describe such a measure on a configuration space that is associated with SAs finite-time performance—the entropy of the Markov chain embodying SA. This connection is rooted in information theory and derives from modeling SA as an *information source.* They show theoretically that for a given instance of a COP, the higher this entropy measure, the better the expected objective function value upon termination of the algorithm. The theory, however, does not quantify the degree to which this relationship holds.

This article therefore serves three purposes. One purpose is to develop practical results and guidelines to improve SAs performance; e.g., to identify ways to increase the entropy measure. The second purpose is to present the development of a methodology and empirical approach designed to explore a particular theoretical relationship. The third purpose is to highlight the connection between thermodynamic/statistical mechanics entropy and information theoretic entropy. The SA algorithm is well-suited for exploring this connection because SAs foundation lies in thermodynamics, and SAs implementation can be modeled as a Markov information source.

The article is organized as follows. Section 1 provides a brief overview of the theoretical results presented in [5]. Section 2 describes the experimental methodology and implementation issues associated with SA. Section 3 presents computational results that illustrate the relationships described in Section 1. Section 4 provides a summary and conclusion of the research presented.

1. Theoretical Foundations

Fleischer and Jacobson[5] present the theoretical foundations of the empirical approach discussed in Section 2. These foundations are based on modeling the SA algorithm as an

Subject classifications: Logic modeling and computation.
Other key words: Information theory, simulated annealing, entropy.

inhomogeneous Markov information source. Such sources can be modeled as an inhomogeneous Markov chain, hence allow certain information theoretic concepts, such as the Asymptotic Equipartition Property (AEP) for ergodic information sources [8, p. 44] to be associated with SA.

The AEP describes the statistical characteristics of a Markov information source. In the context of SA, this provides insights into the asymptotic convergence (in probability) to global optima. This suggests the following question: What are the features of SA *when viewed as an information source* that cause it to converge faster and, consequently, yield better solutions in fewer iterations? To this end, the essential elements of the AEP must be addressed.

1.1 The Asymptotic Equipartition Property

The AEP, as applied to ergodic homogeneous Markov information sources (mathematically modeled as a homogeneous Markov chain; see [8]), asserts that finite sequences of symbols (states) generated by such sources can be partitioned into two mutually exclusive and exhaustive sets: a *typical set*, and an *atypical set*. This partitioning is such that the total probability of the typical sequences can be made arbitrarily close to one as the length of these sequences increases. The AEP further states that the size of the typical set (the number of sequences in it) is dependent on the entropy of the Markov information source that generates it—*the higher the entropy, the larger the size of the typical set*. The AEP therefore associates the size of the typical set and the total probability of the typical sequences with a scalar quantity, the entropy of the Markov chain [8, p. 68].

Ergodic information sources are modeled using aperiodic, irreducible, homogeneous Markov chains, and thus provide the framework of the AEP. Such information sources generate sequences of symbols (or states) where the relative frequencies of symbols *within a sequence* are the same as the relative frequencies *among the typical sequences*. Thus, with enough symbols generated, "for every pattern of output the source can produce, it *will* do so, with asymptotically the right frequency. The effect of the initial conditions dies out, in a strong sense" [8, p. 68] (see also [14, p. 16]). This means that in a typical sequence, the relative frequency of a particular state will be approximately the same as the proportion of typical sequences with that particular state as its final state.

This consequence of ergodicity provides a clue on how entropy can be connected to the finite-time performance of SA. Afterall, SA experiments can be described as strongly ergodic, inhomogeneous Markov information sources.[5] When SA converges (in probability), SA experiments generate a sequence of states (solutions) i.e., a "pattern of output" such that the frequency of visits to globally optimal states increase. Therefore, when SA is run for a sufficiently large number of iterations, the likelihood that the final state is a global optimum increases. Thus, sequences with *high numbers of optimal states (solutions) at the end of the sequence* can be considered typical. When an SA experiment visits states in an unusual manner, such as when it *never* visits optimal states, such a sequence of states has a low probability of occurrence, hence it can be considered atypical. This dichotomy suggests the possibility of extending the AEP to SA experiments, thereby relating typical sequences to the number and probability of such sequences *and the entropy of the associated Markov chain*.[5]

1.2 Expressing the AEP

The analogy between the SA algorithm and Markov information sources requires that the SA algorithm be applied to discrete problems such as intractable COPs. All COPs have a finite number s of states (solutions) that comprise the state (solution) space. Each state $i \in \{1, 2, \ldots, s\}$ has an objective function value f_i and a set of neighboring states, $N(i)$, that defines the *neighborhood structure*. The state space, together with the objective function values and neighborhood structure, constitutes the *configuration space* for the COP. The configuration space, together with the cooling schedule for SA, $\{t_k\}$, (described in [19]) defines an inhomogeneous Markov chain that models the execution of the SA algorithm on a COP.

The following definitions are needed to present the results. For candidate solutions (see [5, 19] for a full exposition of the SA algorithm) generated uniformly over all neighboring solutions, the transition probabilities that an SA experiment moves from state i to state j at time index $k \in Z^+$ (hence at temperature t_k) is given by

$$p_{ij}^{[k]} \equiv p_{ij}(t_k) = \begin{cases} \dfrac{1}{|N(i)|} e^{-\Delta f_{ji}^{+}/t_k} & j \in N(i),\ j \neq i \\ 1 - \displaystyle\sum_{\substack{l=1 \\ l \neq i}}^{s} p_{il}(t_k) & j = i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where

$$t_k = \frac{\gamma}{\log(c + k)} \tag{2}$$

ensures convergence in probability of SA[19] and $\Delta f_{ji}^{+} \equiv \max\{0, f_j - f_i\}$. Note, that the time index k is dropped when the transition probabilities are constant, i.e., in a homogeneous Markov chain. The following definitions are also needed:

- $v_i^{[k]}$, the probability that an SA experiment is in state i at time index k.
- $\pi_i(t)$, the stationary probability of state i at fixed temperature t. The temperature t is dropped when the transition probabilities are constant, i.e., in a homogeneous Markov chain.
- z_k, a state at time index k.
- $Z_n \equiv z_0 z_1 \ldots, z_{n-1}$, a random sequence of n states from time index 0 to $n - 1$.
- $\Pr\{Z_n\} \equiv v_{z_0}^{[0]} p_{z_0 z_1}^{[0]} p_{z_1 z_2}^{[1]} \cdots p_{z_{n-2} z_{n-1}}^{[n-2]}$.
- m_{ij}, the number of times the state pair ij occurs in a sequence of states.
- $H \equiv -\lambda \sum_{i=1}^{s} \sum_{j=1}^{s} \pi_i p_{ij} \ln p_{ij}$, the entropy of a *homogeneous* Markov chain.

$H(k) \equiv -\lambda \sum_{i=1}^{s}\sum_{j=1}^{s} v_i^{[k]} p_{ij}^{[k]} \ln p_{ij}^{[k]}$, the entropy of an inhomogeneous Markov chain at time index k.

$H(m, n-2) \equiv \sum_{k=m}^{n-2} H(k)$, the sum of the entropies of an inhomogeneous Markov chain from time index m to $n-2$.

There are two methods of stating the AEP for ergodic, homogeneous Markov information sources (see [8]).

Method 1.
The frequency of state pairs asymptotically approaches their expected values, i.e., $m_{ij}/n \overset{a.s.}{\to} \pi_i p_{ij}$ as $n \to +\infty$.

Method 2.
A function of the probability of a sequence asymptotically approaches the entropy of the ergodic, homogeneous Markov information source i.e.,

$$-\ln \Pr\{Z_n\}/n \overset{L^2}{\to} H \text{ as } n \to +\infty.$$

Both methods can be used to define typical and atypical (finite) sequences (described in detail in [5]). For finite, ergodic, homogeneous Markov information sources these methods are mathematically equivalent, i.e., one implies the other (see e.g., [8, 14]). For *finite*, ergodic, *inhomogeneous* Markov information sources, such as those in SA, this equivalence does not hold (it holds asymptotically,[5] hence the emphasis above on the word *finite*) because the probability measure of a typical sequence is affected by the time indices of the states.[5] Consequently, differences exist between how the two methods define typical sequences.

These differences raise methodological problems for associating the finite-time performance of SA with entropy. The finite-time performance of SA can be measured by the difference between the final state vector and the optimal state vector. This difference can be estimated by the number of sequences that end in an optimal state, implying that Method 1 must be used to determine typical status. On the other hand, relating this value to entropy involves the probability of the typical sequences (Method 2). Unfortunately, the probability of two distinct typical sequences with identical frequencies of states will be different due to the dependence on the time indices of the states. Thus, performance can be related, directly, only to *entropy-like* measures *different for each typical sequence* (as opposed to a single entropy measure as in the homogeneous case). Note that these entropy-like measures are more difficult to compute than the entropy of the underlying Markov chain.[5, 10]

To address these computational difficulties, Fleischer and Jacobson[5] developed order-of-magnitude estimates of how the expected objective function value is related to the entropy of the Markov chain. The following empirical methodology was developed that illustrates this relationship.

2. Empirical Methodology

This section describes the foundation of the empirical approach and its implementation through the use of *generic configuration spaces* (GCSs), transformation algorithms, and penalty parameters. The basis for this methodology is affected by several empirical constraints. These constraints are imposed by the goal of showing an entropy/performance connection and certain attributes of information sources.

2.1 Empirical Constraints

The AEP provides the empirical constraints by virtue of three aspects of information sources, namely

- sequences of states of finite length,
- the entropy of the associated inhomogeneous Markov chain, and
- the characterization of the sequences, i.e., whether the sequences are typical or atypical.

The first empirical constraint is suggested by the first item, namely, that to assess the relative performance of SA on two (or more) distinct problem instances, SA experiments should use the same number of iterations. This empirical constraint provides the algorithm with a "level playing field."

The second and third items concern more complex aspects of SA as an information source. One useful approach in exploring whether and what type of changes in performance are associated with changes in entropy is to apply SA to different problem instances. In this way, the relative change in performance between two problem instances can be compared to the relative change in entropy measures. For this approach to be effective, however, these measures must be directly attributable to the problem instances themselves and not to the particular implementation of the SA algorithm on any one problem instance.

Recall that the application of the SA algorithm to a given problem instance gives rise to transition probabilities (Eq. 1) and state probabilities $v_i^{[k]}$ that define the entropy of the inhomogeneous Markov chain.[5] Thus, the entropy depends on these probabilities as does the performance of SA (it is the transition probabilities that cause SA to converge in probability to the global optima). These in turn depend on various elements associated with problem instances (the neighborhood structure and the objective function values) and also on SA implementations, i.e., the cooling schedule. All these factors weigh in to affect the entropy and performance measures. Thus, to limit the effects on entropy and performance measures in these comparisons *to the attributes of the problem instance themselves* requires that the cooling schedule be common to all SA implementations. Finally, to make performance differences among different problems readily apparent, the globally optimal objective function values in the different problem instances must be the same.

Putting all these considerations together leads to the following three principles for the experimental methodology used here.

I. SA must be run on various problem instances using the same cooling schedule.
II. Problem instances must have the same globally optimal objective function value.
III. Problem instances must be different from each other (except for Principle II) so that there are differences in entropy and performance. These differences can be established in two fundamentally different ways:

i. configuration spaces can be different yet *functionally related*, or
ii. they can be completely unrelated, i.e., distinct and have no functional relationship.

A functional relationship between two configuration spaces means there exists a mechanism by which one configuration space can be completely determined from another configuration space. If no functional relationship exists, then it is impossible to determine one configuration space from another.

The possibility exists that any association between entropy and performance depends on some explicit or hidden relationship between two configuration spaces. Attributes associated with one configuration space, e.g., the neighborhood size and objective function values, can impact such values in a functionally related configuration space. Thus, the *ways* in which configuration spaces can be different from one another needs to be explored; the concepts of distinct and functionally related configuration spaces provides one mechanism by which to study these differences.

These three principles form the core of the empirical methodology and are incorporated into three different methods for creating and comparing problem instances. The first method uses *generic configuration spaces* (GCSs).

2.2 Generic Configuration Spaces

To implement the three principles, SA must be applied to various problem instances. To this end, it is useful to develop a *general* configuration space. One convenient device for accomplishing this is through the use of GCSs.

The term *generic* is used because GCSs do *not* depend on a *particular* COP. GCSs are sufficiently general to model *any* COP, i.e., the solution space, the objective function values and the neighborhood structure. Given these elements, any arbitrary COP instance can be modeled as a particular GCS instance to which SA is applied.

In GCSs, the solution space is created by randomly generating a list of objective function values and superimposing (on these values) a neighborhood structure. Note that specifying all possible objective function values is an inefficient method of encoding a COP and is, in some sense, computationally equivalent to exhaustive search (making SA unnecessary). So why use GCSs?

Specifying the objective function values and neighborhood structure provides total flexibility in setting the size and topology of the configuration space. For instance, GCSs can be made bumpy (by setting wide differences between neighboring objective function values) or smooth (by setting narrow differences between neighboring objective function values), and the neighborhood sizes can easily be changed. How efficiently a configuration space has been encoded is not a concern—the goal here is to explore the entropy/performance connection in SA and GCSs offer a very broad domain in which to explore this connection.

2.2.1 Families of Generic Configuration Spaces

To create GCSs that conform to Principle III, two distinct solution spaces must be created so that each yields different entropy and performance measures. A simple way to achieve this is to use different ranges of objective function values. Consequently, two solution spaces are created using two distinct uniform distributions. This can have a significant effect on the entropy measure as Figure 1 illustrates.

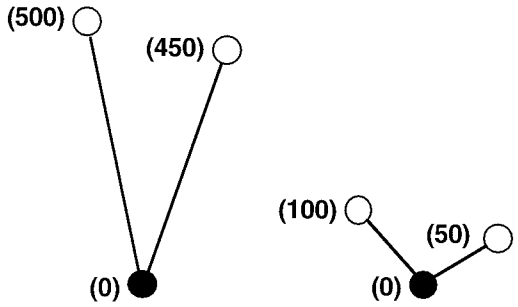

Figure 1. The effect of topology on entropy.

Figure 1*a* depicts an optimal solution connected to two high-cost neighbors, while Figure 1*b* depicts the optimal solution connected to two low-cost neighbors. The row entropy measure (the entropy of the state corresponding to a row in the Markov transition matrix) for the optimal solution is low in Figure 1*a* and high in Figure 1*b*. Deep pits in the configuration space are associated with low row entropy values because there is less uncertainty about the next state—it is more likely that the current state will be the next state because of the depth of these pits. A smoother topography on the other hand is associated with higher entropy measures as there is greater uncertainty regarding the next state.[4]

Each distinct solution space gives rise to a family of configuration spaces that are all functionally related. This functional relationship exists by virtue of the underlying solution space as described in the next section.

2.2.2 Functionally Related Generic Configuration Spaces

To create functionally related GCSs, different neighborhood sizes are used on a given solution space. Let f_i be the i^{th} objective function value associated with solution S_i in solution space $S \equiv \{S_1, S_2, \ldots, S_s\}$. Define $N(i)$, the set of neighbors of S_i by

$$N(i) \equiv \left\{ S_j : \left(i - \frac{N}{2} \right) \bmod s \leq j \leq \left(i + \frac{N}{2} \right) \bmod s,\ j \neq i \right\} \tag{3}$$

where N is the neighborhood size. To illustrate, for $N = 2$, the neighbors of S_i are S_{i+1} and S_{i-1}. Note that the modulus allows the neighbors of solutions 1 and s to wrap around such that S_s and S_1 are adjacent in the solution space. Thus, the neighbors of S_s are S_1 and S_{s-1}, while the neighbors of S_1 are S_s, and S_2. Also, Eq. 3 creates neighborhood sizes that are multiples of two. This avoids the complications associated with odd neighborhood sizes and simplifies the computation of the entropy and performance measures (by giving a structure to the Markov transition matrices in SA; see Section 3).

Equation 3 is sufficient to define numerous GCSs based on a single solution space. Consequently, such GCSs are functionally related by Eq. 3. In addition, Eq. 3 makes it easy to change the neighborhood size *without changing the globally optimal value* (Principle II). GCSs with different neighborhood sizes can have significant topological differences (notwithstanding their common solution space), hence directly affect the structure of the Markov transition matrices, the values of the transition probabilities, the entropy measures, and the performance of SA.[4]

The approach of using two distinct solution spaces is an efficient and flexible method for creating test-bed problems that conform to the three principles defining the empirical methodology. This approach establishes two distinct *families* of configuration spaces. Any GCSs based on the same solution space are therefore functionally related. To adhere to Principle II, the globally optimal values in each solution space are set to zero. This implements all three principles, which are now applied to actual COPs.

2.3 COPs and Transformation Algorithms

The second approach that implements the three principles is based on polynomial transformations of *NP*-hard COPs. This approach creates changes in the configuration space by transforming (reducing) one COP into another, different COP. By virtue of the *transformation* algorithm, functionally related configuration spaces can be created that adhere to Principle III by altering the entire configuration space (the solution space, the objective function values, and the neighborhood structure). Yet, the transformed problem has the same globally optimal objective function value, thereby adhering to Principle II. Note that the results of the experiments involving polynomial transformations must be studied separately from the experiments using GCSs as the optimal solutions for these sets of experiments are not guaranteed to be the same. To illustrate these ideas, SA algorithms were developed for optimization versions of the clique and the 3SAT problems (see [7, p. 46–47]) because of the simplicity of the algorithm that transforms a 3SAT problem instance into a clique problem instance. As in the case of GCSs, two distinct families of 3SAT-Clique problem instances were developed. These problem instances are discussed in more detail in the following sections.

2.3.1 The 3SAT Problem

In the optimization version of the 3SAT problem, the objective is to determine the truth assignment for a set of Boolean variables that maximizes the number of Boolean clauses that are true, where each clause has exactly three variables (see [7, p. 259]). Thus, given a truth assignment $U = \{u_1, u_2, \ldots, u_n\}$ over a set of clauses C, define the objective function

$$f_{3\text{SAT}}(U, C) = \sum_{c \in C}^{|C|} c$$

where the truth value of clauses $c \in C$ are determined by the truth assignment U. The size of the configuration space for $n = |U|$ Boolean variables is 2^n. As in the clique problem, each solution can be perturbed by flipping or inverting the truth value of one Boolean variable. Consequently, each solution also has n neighbors.

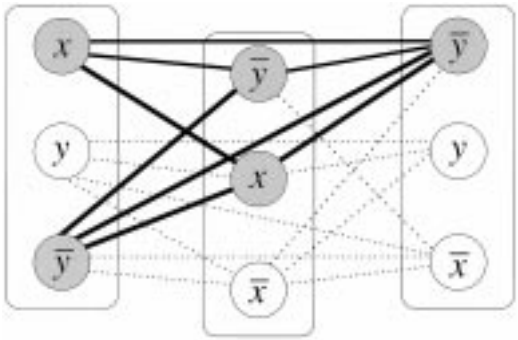

Figure 2. The 3SAT-Clique transformation.

2.3.2 The Maximum Clique Problem

Given a graph $G = (V, E)$, with a set of vertices V and a set of edges E with $V' \subseteq V$, define

$E(V')$, the number of edges not connecting nodes in set V'.
i.e., $\forall u, v \in V', u, v \notin E$,
$v' = |V'|$, the number of nodes in V'.

Define the objective function as described in [1, p. 81]:

$$f_{\text{clique}}(V', G) = v' - \alpha E(V') \tag{4}$$

where V' is the set of nodes corresponding to the current solution, α is a penalty parameter, and $E(V')$ is the number of missing edges in V' that violate the constraints of the clique problem.

Note that because this is a maximization problem, the penalty term is subtracted from the number of nodes. A consequence of using penalty parameters is that all combinations of nodes become feasible solutions in the solution space. Candidate solutions can therefore be generated simply by adding or subtracting (flipping) any node from a current solution. Moreover, every solution has the same neighborhood size. Note that this yields a solution space of size 2^n, where n is the number of nodes in the graph. For additional details on this problem, see [3, 7].

2.3.3 Transformation Algorithms

The theory of *NP*-completeness indicates that all *NP*-hard COPs can be related by polynomial transformation algorithms. Thus, an algorithm exists that transforms a 3SAT problem instance into a clique problem instance in which the optimal solutions for both problems are the same. By taking advantage of this principle, the configuration space can be altered, adhering to Principle III, without changing the optimal objective function value, adhering to Principle II. The transformation from a 3SAT problem instance to a clique problem instance therefore constitutes a way of functionally relating two problems instances. For details of this transformation see [16, p. 352]). Figure 2 illustrates this transformation. The rectangular boxes correspond to clauses, each with

three Boolean variables from a set of two Boolean variables ($\{x, y\}$) represented by circles. The circles correspond to nodes in a graph and the lines correspond to the arcs in the graph based on the transformation. The shaded nodes correspond to Boolean variables that are true. Thus, $x = \bar{y} = 1$ and the three clauses are *satisfied*, hence the objective function value for this 3SAT problem is three. Notice that for the Boolean variables that are set to 1, the lines connecting the corresponding nodes are bolded. These bolded lines constitute cliques, each of which is the same size as the number of clauses with truth value 1; four cliques of size three are apparent.

This transformation yields a configuration space for the clique problem significantly different from the original 3SAT problem. In particular, for n Boolean variables in a 3SAT problem instance with $|C|$ clauses, there are 2^n solutions. Transformation to clique changes the number of solutions from 2^n to $2^{3|C|}$ and the neighborhood size from n to $3|C|$. If $|C| > n/3$, then the neighborhood size and the solution space of the clique problem instance both increase relative to the 3SAT problem instance.

2.4 Penalty Parameterization: Another Functional Relationship

Another approach for creating functionally related configuration spaces is to change the penalty parameter α in Eq. 4. Changing the value of α changes the objective function values associated with a given combination of nodes. This, in turn, changes the transition probabilities of the Markov transition matrices and leads to changes in both entropy and performance. Just as Eq. 3 establishes functionally related GCSs with a common solution space, Eq. 4 establishes functionally related configuration spaces for clique problems with a common graph.

2.5 Summary of the Methodology

The three principles are implemented in several ways: by using two distinct solution spaces, two distinct families of GCSs are created. Changing the neighborhood size associated with each solution space produces two sets of functionally related GCSs all adhering to the principles. The theory of *NP*-completeness and polynomial transformation algorithms also provide a way for creating functionally related configuration spaces. Transformation algorithms are applied to two distinct 3SAT problem instances to create two families of functionally related clique problems. Different values of the penalty parameter in the objective function for the clique problem establishes many functionally related configuration spaces. Using these three qualitatively different approaches for producing configuration spaces therefore permit a large number of comparisons to be made between entropy and performance. Computational results using these methods are presented in the next section.

3. Computational Results

This section presents computational results using the empirical methodology described in Section 2. Before describing these results, it is important to note that the computation of the entropy and performance measures is based on vector-matrix multiplications and are thus analytically computed, *not* estimated by a series of simulation experiments. The software used was written in the C programming language and executed on Sun Sparc Workstations and PowerMac computers.

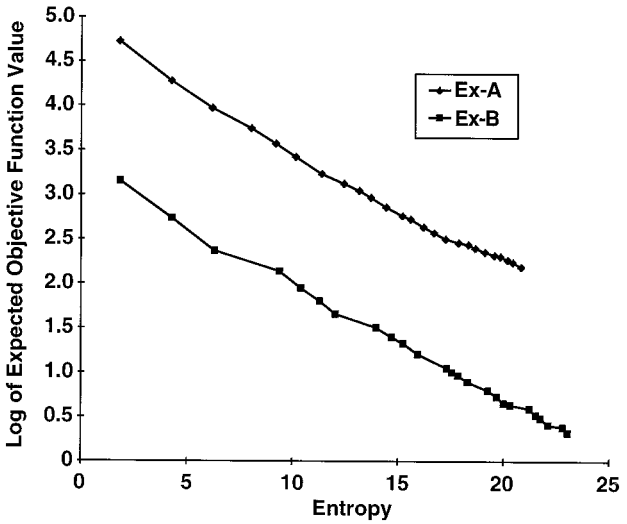

Figure 3. Log of expected objective function value versus entropy for GCSs.

3.1 Generic Experiments

Two families of GCSs were created using two distinct solution spaces each with 5000 solutions with integer objective function values in the following ranges:

GCS A {0 to 500},
GCS B {0 to 100}.

Computations for GCSs are based on 100 iterations of SA with cooling schedule (Eq. 2), an initial temperature of 10,000, and a final temperature of 0.1 (see [5] for a description of how a fixed-point theorem was used to establish parameters in the cooling schedule). The initial temperature value was chosen such that at the first iteration the probability of accepting the worst-case uphill jump is at least a 0.5. For example, for a jump from 0 to 500, an initial temperature of 10,000 yields an acceptance probability of 0.606.

The computational results are presented in Figure 3. The x-axis shows the entropy of the Markov chain. The y-axis shows the logarithm of the expected objective function value based on the final state vector. Points correspond to a given neighborhood size. These neighborhood sizes increase in increments of 2 and range from 2 to 50 for a total of 25 GCSs for each family (points associated with each family of GCSs are connected) yielding a total of 50 GCSs. Each GCS is therefore functionally related to 24 other GCSs and provide for 625 comparisons of entropy and performance measures!

The most significant feature of Figure 3 is that for each such family of GCSs an increase in entropy measure is associated with a decrease in expected objective function value, i.e., an improvement in performance. Moreover, this

association is monotonic *with respect to neighborhood size*. Note, that the log scale and the clustering of points toward the lower *y*-axis values reflects that increases in neighborhood size lead to diminishing improvements in performance and to diminishing increases in entropy measures. Intuitively, GCS A should (and does) have higher *expected* objective function values (due to the higher objective function values) compared to GCS B, hence a greater likelihood of having a larger number of deep pits (see Figure 1).

These two curves also suggest that the entropy/performance relationship is not absolute *with respect to each family of GCSs*. The monotonic nature of both curves and the fact that they are nearly straight lines suggests that the expected objective function value is an exponential function of entropy (see [4, 5]). Moreover, these lines are nearly parallel, further suggesting that the entropy/performance relationship for these distinct families of GCSs are related by a scaling factor.

The conclusion to be drawn from these results is that for *functionally related GCSs, higher entropy Markov chains are associated with superior SA performance*. Moreover, an effective way to increase the entropy is to increase the neighborhood size. Further research may shed more light on whether distinct families of configuration spaces can, indeed, be related by a scaling factor.

3.2 Combinatorial Optimization Problem Experiments

This section describes the results obtained from applying SA to a 3SAT problem instance and its functionally related clique problem instance. To discern entropy/performance differences, the number of iterations for these problems was kept small to avoid convergence to the globally optimal values. To do otherwise would produce nearly the same solutions in both problems, thereby masking finite-time differences in convergence rates (see [6] for a description of this methodological issue).

3.2.1 The 3SAT/Clique Problems

Conforming to the methodology used for the GCS experiments, two distinct 3SAT problem instances were created, denoted as 3SAT-A and 3SAT-B, using 11 and 12 Boolean variables, respectively.[12] Each was comprised of five randomly generated clauses in which the optimal solutions for both were the same (Principle II), i.e., 5. The configuration space for 3SAT-A has $2^{11} = 2048$ solutions each with 11 neighbors; 3SAT-B has $2^{12} = 4096$ solutions each with 12 neighbors. Both 3SAT problem instances were transformed to ten clique problem instances (Clique A, Clique B) each comprised of 15 vertices corresponding to a solution space size of $2^{15} = 32768$ each with 15 neighbors.

The SA algorithm was applied to these problem instances in the analytical fashion noted earlier, namely, a series of vector-matrix multiplications using identical cooling schedules. The initial temperature was 400. The cooling schedule is such that the temperature at iteration 50 is 0.1, although only 10 iterations were computed. Figure 4 shows the expected objective function and entropy measures associated with both 3SAT problems (the marker $\times$ for 3SAT-A, the marker + for 3SAT-B), and their functionally related clique problems are described in greater detail below. The effect on entropy and performance of the transformation algorithm is indicated with the dashed lines. Notice that the transformations yield points that are shifted to the left (lower entropy) and down (worse performance; both instances are maximization problems).

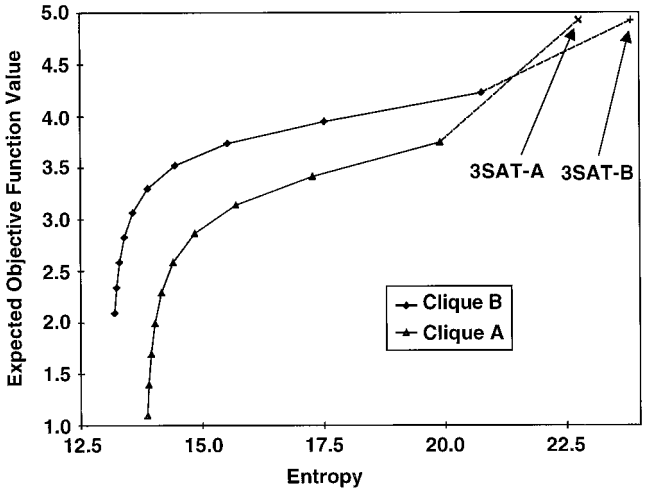

Figure 4. Expected objective function versus entropy for 3SAT-Clique.

3.2.2 Changes in the Penalty Parameter

To illustrate the entropy/performance connection using changes to the penalty parameter in the clique, two families of clique problem instances were derived from the 3SAT problem instances. For each clique problem, ten configuration spaces were created using ten values of the penalty parameter α. Increases in α exaggerate the peaks and valleys in the configuration space. The basic structure of the configuration space stays the same but with different relief. Consequently, all of these configuration spaces are functionally related by Eq. 4. The $\triangle$ and $\diamond$ markers in Figure 4 show the entropy/performance points for Clique A and B, respectively. Functionally related clique problem instances are connected by a line.

Figure 4 depicts a similar relationship between entropy and expected objective function values as implied in Figure 3 with diminishing improvements in expected objective function values for the larger entropy values. Note that the entropy and the expected objective function values are both monotonic with respect to the penalty parameter α (each point going from left to right varies from 2.9 to 1.1 in increments of 0.2). As the parameter decreases, both the entropy measures and the expected objective function value increase. One other interesting aspect is that the points corresponding to each value of α in Clique B have higher expected objective function values than the corresponding points in Clique A, reflecting the effects of a larger neighborhood size of its parent 3SAT problem (12 versus 11).

The relationship between the two families cannot be as

clearly depicted as in Figure 3 as it was necessary to use fewer iterations. Nonetheless, it is worth noting that for both families, the entropy / performance relationship is very similar.

4. Summary and Conclusions

A connection, based on information theory, between the entropy of an inhomogeneous Markov chain and the convergence of that Markov chain (as measured by the finite-time performance of SA) was articulated. Three principles for applying an empirical methodology were established to explore this connection, and three methods were developed that incorporate these three principles.

The first method was based on the concept of GCSs. Two families of functionally related GCSs showed that increases in neighborhood size increased the entropy values and improved the finite-time performance of SA. Moreover, a positive relationship between entropy values and performance was demonstrated. Comparisons between distinct families of GCSs suggest that the entropy / performance connection can be related by a scaling factor.

A second method based on COPs and transformation algorithms confirms the results obtained with the functionally related GCSs. Two distinct 3SAT problem instances were transformed to clique problem instances using a polynomial transformation algorithm, thereby creating two functionally related configuration spaces. The transformed 3SAT problems, i.e., the clique problems, both showed decreases in entropy values and decreases in performance.

Finally, a third method, based on penalty parameterization of the clique problems, was used to create two families of functionally related configuration spaces. Again, changes in entropy values matched changes in performance.

All these data support the theory that the expected value of the final state in an SA experiment is related to the entropy of the Markov chain and that this entropy value determines how well SA performs. For both minimization problems (the GCS experiments) and maximization problems (the COP experiments), higher entropy values were associated with improved expected objective function values.

The theories and experimental methodologies developed not only answer questions regarding the performance of SA, but also provoke other questions. For example, the experiments show that increases in entropy can generally be achieved by increasing the neighborhood size (as suggested by the GCS experiments), and / or smoothing out the topology of the configuration space (to the extent it does, in fact, increase the entropy) as suggested by the penalty parameter experiments. Consequently, researchers can explore computationally efficient ways to increase the neighborhood size or develop ways to smooth out a configuration space.

The experiments also indicate that there is some aspect regarding functional relationships between problem instances that constrains both the performance and entropy in similar ways. This may also be worth further exploration. Answers to such questions may provide information on the "optimal" way to modify a problem so as to improve the effectiveness of SA. Is it possible that there is such a thing as an *optimal problem instance*? At a minimum, future research may make it possible to define a certain class of problems or transformation algorithms that tend to be associated with high entropy values. Conceivably, this can save practitioners time by enabling them to decide *a priori* whether SA is an appropriate method to use.

Of greater significance is the fact that COPs can all be viewed from an information theoretic and thermodynamic perspective using a common and consistent framework from which comparisons can be made—the SA algorithm. In this way, COP instances can be assigned an information measure. For instance, one COP may be *informationally strong* (high entropy measure) and another, informationally weak (low entropy measure). Ways to assess just how "hard" an *NP*-hard problem is may also be illuminated. At present, the theory of *NP*-hardness classifies COPs using the fact that one problem is *at least as hard as* some other COP.[7] This classification is based on worst-case analysis, and polynomial complexity, hence, can be indiscriminate insofar as qualifying specific COPs. But the notion of average-case analysis, an alternative approach seen by some as a more realistic classification scheme, suffers from many methodological problems, not the least of which is defining an appropriate sample space (see [10]). The information theoretic approach in the context of SA avoids some of these difficulties. This may provide a basis for measuring the information content associated with COPs and, further, provide a new way of classifying them and measuring the difficulty for solving them.

The paradigm of viewing COPs as information-measurable entities provides a new avenue that can be exploited by future researchers. The idea that a connection exists between the information produced by an algorithm (information-theoretic entropy), the information content of a problem (statistical-mechanics entropy), and an algorithm founded on thermodynamic principles (SA) is a compelling one. In other contexts, this connection has been described using Maxwell's Demon, the imaginary creature who persistently attempts to violate the second law of thermodynamics by obtaining information about a system without paying the necessary price in terms of energy. As Pierce notes: "One pays a price for information which leads to a reduction of the statistical-mechanical entropy of a system. This price is proportional to the communication-theory entropy of the message source which produces the information" [22, p. 206]. SA simulates the reduction of the energy level of a thermodynamic system. The more information produced by an SA experiment, the greater is this reduction in "energy," hence, the closer the simulation will be to the "ground state," i.e., the optimal solution to a COP.

Acknowledgements

The authors would like to thank the area editor John N. Hooker and three anonymous referees for their insightful observations and recommendations, which have lead to a significantly improved document. The first author was supported in part by a NASA contract (NAS1-19858-13). The second author was supported in part by grants from the NSF (DMI-9409266, DMI-9423929) and the AFOSR (F49620-95-1-0124, F49620-98-1-0111).

References

1. E. AARTS and J. KORST, 1989. *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*, John Wiley & Sons, New York.
2. R.S. BARR, B. GOLDEN, J.P. KELLY, M.G.C. RESENDE, and W.R. STEWART, JR., 1996. Designing and Reporting on Computational Experiments with Heuristic Methods, *Journal of Heuristics 1*, 9–32.
3. J.A. BONDY and U.S.R. MURTY, 1976. *Graph Theory with Applications*, North-Holland, Elsevier Science Publishing Co., Inc., New York.
4. M. FLEISCHER, 1993. Assessing the Performance of the Simulated Annealing Algorithm Using Information Theory, Doctoral dissertation, Case Western Reserve University, Cleveland, OH.
5. M. FLEISCHER and S.H. JACOBSON, 1998. Assessing the Finite-Time Performance of the Simulated Annealing Algorithm Using Information Theory. Technical Report, Old Dominion University, Norfolk, VA.
6. M. FLEISCHER, 1998. Cybernetic Optimization by Simulated Annealing: Solving Continuous Variable Problems. *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, Ch. 28*, Kluwer Academic Publishers, Norwell, MA, 403–418.
7. M. GAREY and D.S. JOHNSON, 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Co., New York.
8. C. GOLDIE and R. PINCH, 1991. *Communication Theory*, Cambridge University Press, New York.
9. L. GOLDSTEIN and M. WATERMAN, 1988. Neighborhood Size in the Simulated Annealing Algorithm, *American Journal of Mathematical and Management Sciences 8*, 409–423.
10. J.N. HOOKER, 1993. Needed: An Empirical Science of Algorithms, *Operations Research 42*, 201–212.
11. J.N. HOOKER, 1995. Testing Heuristics: We Have It All Wrong, *Journal of Heuristics 1*, 33–42.
12. J.N. HOOKER, 1993. Correspondence.
13. D.S. JOHNSON, C.R. ARAGON, L.A. MCGEOCH, and C. SCHEVON, 1989. Optimization by Simulated Annealing: An Experimental Evaluation, Part I (Graph Partitioning), *Operations Research 37*, 865–892.
14. A.I. KHINCHIN, A.I., 1957. *Mathematical Foundations of Information Theory*, Dover Publications, Inc., New York.
15. P. L'ECUYER, 1996. Simulation of Algorithms for Performance Analysis, *INFORMS Journal on Computing 8*, 16–20.
16. U. MANBER, 1989. *Introduction to Algorithms: A Creative Approach*, Addison-Wesley Publishing Company, New York.
17. C.C. MCGEOCH, 1996. Toward an Experimental Method for Algorithm Simulation, *INFORMS Journal on Computing 8*, 1–15.
18. C.C. MCGEOCH, 1996. Challenges in Algorithm Simulation, *INFORMS Journal on Computing 8*, 27–28.
19. D.F. MITRA, D., F. ROMEO, and A. SANGIOVANNI-VINCENTELLI, 1986. Convergence and Finite-Time Behavior of Simulated Annealing, *Advances in Applied Probability 18*, 747–771.
20. J.B. ORLIN, 1996. On Experimental Methods for Algorithm Simulation, *INFORMS Journal on Computing 8*, 21–23.
21. R.H.J.M. OTTEN and L.P.P.P. VAN GINNEKEN, 1989. *The Annealing Algorithm*, Kluwer Academic Publishers, Norwell, MA.
22. J.R. PIERCE, 1980. *An Introduction to Information Theory: Symbols, Signals and Noise, 2nd Ed.*, Dover Publications, Inc., New York.
23. D.R. SHIER, 1996. On Algorithm Analysis, *INFORMS Journal on Computing 8*, 24–26.