Simulated Annealing: A Heuristic for Parallel Stochastic Optimization

submitted to proceedings of PDPTA '97. Copyright 1997 CSREA.

R. Frost
San Diego Supercomputer Center
San Diego, CA, U.S.A.

P. Heineman
San Diego State University
San Diego, CA, U.S.A.

Abstract *An overview of physical annealing and simulated annealing methods is presented. The target audience includes computational analysts and software engineers who are considering simulated annealing for the solution of a combinatorical optimization problem. Several important implementation issues are discussed including decompositions for parallel programming. Special emphasis is given to the often ignored perspective of thermodynamics.*

Keywords: parallel, optimization, stochastic

1 Introduction

Annealing is a formal term for the ancient art of heating and/or cooling materials to forge pottery, tools, weapons, and works of art. It is the process of subjecting a substance to changes in pressure or temperature to achieve desired material properties. Annealing has played a key role in human history: entire nations have been won and lost on the abilities of craftsmen to produce fit materials.

A contemporary example of annealing is the manufacture of glass products from raw silica. The silica is first heated from an ordinary solid phase to a liquid. The liquid silica is then cooled according to a precise *cooling schedule* to achieve a desired solid phase; for example, a solid with better optical properties [1, 2].

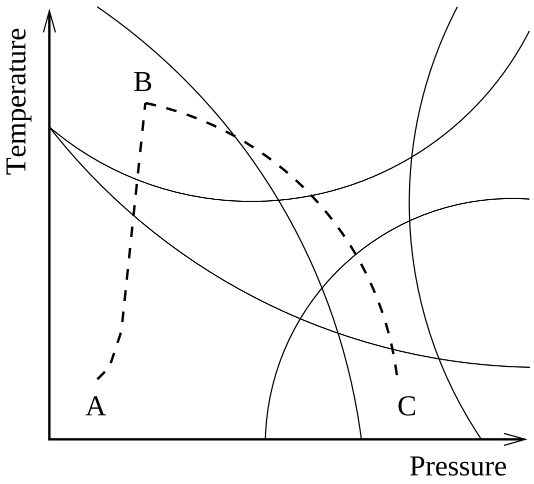

Figure 1: Example temperature-pressure phases of a material. The dotted line shows an annealing path from phase A through phase B to phase C.

A depiction of an annealing process is given in figure 1. Note that the path from A through B to C is not the shortest path in phase space from A to C, but may nonetheless represent the most time and cost efficient path. Although it might be desirable to move directly from A to C, it is sometimes the case that a direct path is too costly, too time-intensive, or physically intractable.

Modern research in annealing techniques includes both the "art" of experimental material science [3, 4] and theoretical work based on statistical thermodynamics [5, 6]. The latter describes the relationship between *macrostates* of the material and the *states* of the atoms or molecules that compose it. A practical application of this theory is to determine what annealing schedule should be observed to effect desired changes in the properties of a given material.

1.1 Fundamentals of Temperature

A material is composed of a set S of atoms or molecules. The *states* $\omega \in \Omega$ of atoms or molecules in S include measures of their position, orientation, rotation, velocity, etc. The energy $E(\omega)$ of any particular atom or molecule is a function of its state at some instant in time. The set of $E(\omega)$'s can be expressed by $\mathcal{E} = \{E(\omega) : \omega \in \Omega\}$. In real materials these energies occur in quantized intervals[1] so that $\#\mathcal{E} \ll \#\Omega$.

A material is said to be at temperature $\Theta(\Omega) = T$ when the individual energies $E(\omega)$ are Boltzmann-distributed at temperature T. For example, the informal statement "the room temperature is 68°F" is equivalent to "the individual energies of the gas molecules in this room are Boltzmann-distributed at temperature $T = 293°\mathrm{K}$". For systems of non-interacting particles, this can be mathematically stated:

$$\begin{aligned} \Theta(\Omega) = T \;\; &\Leftrightarrow \;\; \forall\, \omega \in \Omega\,, \\ p(\omega \,|\, T) \;\; &= \;\; \frac{e^{-E(\omega)/kT}}{\sum_{\omega\in\Omega} e^{-E(\omega)/kT}}\,. \qquad (1) \end{aligned}$$

The Boltzmann constant k may be set equal to 1 by using the same units for energy and temperature.

In real materials, changes in temperature are always due to adding or subtracting energy through conductive, convective, mechanical, or radiative means. It is possible for the energy distribution of a set of atoms or molecules to be in a transition state where temperature is not defined (i.e., not Boltzmann distributed). A system composed of ice cubes dropped into a hot drink are one example.

To extend equation 1 to systems of interacting particles, Gibbs [5] considered an ensemble $\mathcal{G}$ composed of many copies of S. Each $\gamma \in \mathcal{G}$ represents one possible configuration of states $\{\omega\}$ attainable by S. Each γ has an associated bulk state u; e.g., one glass state vs. another. The entire collection of these states $\mathcal{X}$ enumerate the combinatorally large number of possible bulk states available to S. The elements $u \in \mathcal{X}$ are termed *microstates*. In this theoretical framework, the probability of attaining some microstate u is:

$$p(u \,|\, T) \;\; = \;\; \frac{e^{-E(u)/T}}{\sum_{u\in\mathcal{X}} e^{-E(u)/T}}\,. \qquad (2)$$

For example, the microstate energies which have high probability in material phase A of figure 1 will have low probability in phase C. This is true even when states in each phase are equilibrated at the same temperature T. The range of attainable $u \in \mathcal{X}$ are physically constrained by the properties of the constituent atoms or molecules and laws of energy conservation. Microstate statistics for a given material can be obtained by laboratory measurements or derived from theoretical predictions of desirable material phases.

1.2 Pressure

Pressure is a function of constituent states and interaction forces. Changes in temperature will coincide with proportional changes in pressure (and vis a vis) unless volume is increased or diminished to hold the current pressure constant. For example, when a piece of metal is heated in open air, it expands.

1.3 Annealing with Optimal Speed

In the example of figure 1, an initial heating process is required in the transition from phase A to phase C. This is equivalent to acquiring a distribution of microstates associated with phase B for which subsequent transition probabilities to microstates in C are of suitable magnitude. Once a distribution is obtained for which paths to C are tractable, a cooling process is initiated. Of practical importance is how to perform annealing tasks in the most time efficient manner. To establish this result, we will first discuss some basic terminology and

---

[1] We consider only bound states of Ω

constructs, then derive an equation for thermodynamic speed.

In the previous section it was stated that the microstates of material at fixed temperature T will occur with probability $p(u|T)$. Let $P(T)$ represent the distribution of microstate energies at temperature T

$$P(\epsilon,T) \quad \equiv \quad \frac{\#\{E(u)=\epsilon \ \mid \ T\}}{\#\{u\}}\,.$$

If ϵ is notationally suppressed in P, then $P(T)$ represents a vector of fractions indexed by energy vector $\vec{\epsilon}$. The *density of states* ρ is defined as the distribution of P at temperature $T=\infty$. The *heat capacity* of Ω at T is

$$c(T) \quad = \quad \frac{\sigma^2(T)}{T^2}$$

where the standard deviation of the energy distribution is defined

$$\sigma(T) \quad = \quad stdev_\epsilon(P(T))\,.$$

The mean energy $\bar{E}(T)$ is the first moment of the energy distribution

$$\bar{E}(T) \quad = \quad \vec{\epsilon} \oplus P(T)\,.$$

The *transition probability* of moving from configurate u_j to neigboring configuration u_i at infinite temperature is

$$\pi_{i\leftarrow j}(T_\infty) \quad = \quad \frac{1}{d} \tag{3}$$

where d is the number of neighbors of u_i. In otherwords, all states are equally likely at T_∞. At finite temperatures, the probability of *attempted transitions* among neighbors is still equally likely, but with the additional Boltzmann dynamics

$$\tau_{i\leftarrow j}(T) \quad = \quad \begin{cases} 1 & \epsilon_i \leq \epsilon_j \\ e^{-(\epsilon_i-\epsilon_j)/T} & \epsilon_i > \epsilon_j \end{cases}\,. \tag{4}$$

Coupling (3) and (4) yields the finite temperature transition probabilities

$$\pi_{i\leftarrow j}(T) \quad = \quad \begin{cases} \frac{1}{d} & \epsilon_i \leq \epsilon_j \\ \frac{1}{d}e^{-(\epsilon_i-\epsilon_j)/T} & \epsilon_i > \epsilon_j \end{cases}. \tag{5}$$

The Boltzmannized matrix [7] is constructed from π:

$$\mathcal{B}_{ij}(\pi,T) \quad = \quad \begin{cases} \frac{1}{d}\pi_{i\leftarrow j} & i \neq j \\ 1-\frac{1}{d}\sum_{k\neq i}\pi_{i\leftarrow k} & i = j \end{cases} \tag{6}$$

$\mathcal{B}$ is a stochastic matrix with largest eigenvalue 1 corresponding to $p(u|T)$. The second largest eigenvalue λ_2 of $\mathcal{B}$ is related to the thermodynamic relaxation time

$$\zeta(T) \quad = \quad \frac{1}{1-\lambda_2}\,.$$

Knowing $\mathcal{B}$, or an estimate of $\mathcal{B}$, is sufficient for computing a cooling schedule from a known distribution of states to some desired distribution [7]. Complete knowledge of $\mathcal{B}$ is a rare luxury.

The optimal annealing speed to minimize entropy production in a fixed number of equilibrations is

$$\nu \quad = \quad \frac{d\mathcal{L}}{d\xi}$$

where $\mathcal{L}$ is the characteristic length of displacement between distributions and ξ is the characteristic length of time at temperature T [8]. To express $d\mathcal{L}$ as a function of T, we consider

$$\begin{aligned} (d\mathcal{L})^2 \quad &= \quad -\frac{\partial^2 S}{\partial E^2}dE^2 \\ &= \quad -dE^2 \cdot \frac{\partial(\frac{1}{T})}{\partial E} \\ &= \quad dE^2 \cdot \frac{1}{T^2} \cdot \frac{\partial T}{\partial E} \\ &= \quad (c \cdot dT)^2 \cdot \frac{1}{T^2} \cdot \frac{1}{c} \\ &= \quad \left(\frac{dT}{T}\right)^2 \cdot \frac{\sigma^2}{T^2} \\ \Rightarrow \ d\mathcal{L} \quad &= \quad \sigma \cdot \left(\frac{dT}{T^2}\right) \end{aligned} \tag{7}$$

The characteristic time differential is

$$d\xi = \frac{dt}{\zeta}$$

so that

$$\nu \quad = \quad \frac{dT}{dt} \cdot \frac{\zeta(T)\sigma(T)}{T^2}$$

or

$$\frac{dT}{dt} = \nu \cdot \frac{T^2}{\zeta(T)\sigma(T)} \tag{8}$$

In equation 8, $\sigma(T)$ is measurable and $\zeta(T)$ can be computed from (an estimate of) $\mathcal{B}$ or by other thermodynamic considerations. This equation can be numerically integrated for the purpose of simulation. Solutions $T(t)$ are known as constant thermodynamic speed *annealing schedules*. Time-dependent phase relations $< E(T) >$ that result from these solutions are termed *annealing curves*.

2 Annealing as an Optimization Heuristic

Annealing is often cited as a useful heuristic for global optimization problems [9, 10, 11]. Experience has shown that this claim is a bit zealous. In practice, annealing is a poor choice for problems readily solved by standard optimization tools, but is useful when the search space is huge, little is known about the problem structure, or the cost function is noisy with significant variance [12, 13, 14, 15, 16, 17]. The heuristic is constructed as follows:

(Annealing Heuristic)

a. A *value* x in the problem domain D is identified with the microstate of a fictitious material. A set of N values $\{x\}$ in D are identified with the microstate *ensemble* $\{u\} \subset \mathcal{X}$.

b. The cost function $F(x) : x \in D$ is identified with the (fictitious) energy function, $E(u) : u \in \mathcal{X}(\Omega)$.

c. A small perturbation of u (from u to u') is analogous to a small perturbation in the value of x (to x').

d. A change in some u may effect a change in its energy. The transition of u from some energy value $E(u) = \epsilon_m$ to some neighboring energy $E(u') = \epsilon_n$ is identified with a change on the cost surface from $F(x) = f_m$ to $F(x') = f_n$.

e. The reduction of an initial set of values in the problem domain D to a distribution of (nearly) global minima is identified with the cooling of material from some initial temperature T_0 to some final temperature T_f.

The term most often applied to this heuristic is "simulated annealing" (SA). However, it is evident from item **e** above that "simulated cooling" might be a more accurate description. A literal interpretation leads to the following algorithm:

(Algorithm 1)

1. Initial stage:
    - **a.** Initialize N values $x \in D$ to form an ensemble for fictitious material Ω.
    - **b.** Estimate initial temperature T_0.
    - **c.** Equilibrate $\{u\}$ at T_0 by repeated application of the transition probabilities π until $\bar{E}(T)$ approaches some steady state.
2. Cooling stage:
    - **a.** Directly adjust T (according to some schedule) in a generally decreasing trend until some stopping criteria are met; e.g., $\bar{E}(\Omega)$ or some $F(x)$ is sufficiently low.
    - **b.** At each increment of T, equilibrate $\{u\}$ as in **1c**. As T is reduced, the probability of any x making an "uphill" move in cost is also reduced.

In practice, formulating an optimization problem for annealing is easy but achieving satisfying results can be hard. The following two subsections discuss important differences between physical and simulated annealing along with implementation issues. A survey of implementations is presented in section 3.

2.1 Issues and Observations

Physical annealing is an inherently parallel process. In real materials about 10^{23}

atoms or molecules make asynchronous energy transistions in the range of every 10^{-12} to 10^{-9} seconds. Consequently, SA is inherently parallel—albeit on a smaller scale. Historically, SA algorithms have been serialized to accomodate limitations in computer architectures.

At the core of the annealing heuristic is the mapping of individual energy transitions to perturbations in the values of state variables (annealing heuristic, item **d**). A change in the value of any particular x might effect a small or a large change in the value of $F(x)$. This will depend upon the choice of discretization and coordinate systems in a grid structured problem, or the choice of neighborhood regions in an affine or graph structured problem. Such choices are termed *move classes*. If a move class is chosen so that local variances are (almost) everywhere large in comparison with the overall variance of $F(x)$, then the performance of the SA implementation will be extremely poor.

Two useful strategies for selecting a move class [18] are:

- at each time step, the movement of the ensemble as a whole should yield a characteristic value for $|\overline{\Delta E}|$, and
- these ensemble "moves" should be well separated, but within a standard deviation of $\bar{E}$.

Thus, the investigator should strive for a move class that yields

$$\sigma(T) \quad > \quad |\overline{\Delta E}| \; \approx \; T \tag{9}$$

for (at least) readily observable samples of D.

If the size of the search space is exceedingly large, it might be tempting to expand the neighbor-to-neighbor step length to include moves to 2, 3, ... up to L coordinate (or graph) points away from the current state. The motivation here is to shorten the number of steps required to traverse a diameter of D. A typical implementation utilizing this approach selects moves from point j in both a probabilistic direction and a probabilistic fraction of step length L. Doing so increases connectivity and thus increases the dimension of the problem. At some step length, the dimensionality will approach the problem size. Expanding the step length to include all points in the search space is equivalent to random search.

The struggle to find an effective move class can be challenging. The surface of $F(x)$ under any of choice of move class must be largely unknown—otherwise the optimization problem would be (nearly) solved! Nonetheless, it is desirable to construct a move class such that the global cost surface resembles more a set of rolling hills, mountains, and valleys than the characteristic function of a Cantor set. Software tools to assist with the construction of cost functions and move classes would be of great use to researchers.

In SA, a *cooling schedule* refers to the incremental alteration of T in step **2a** of Algorithm 1. Cooling too fast will result in the $\{x\}$ being trapped in local minima, while cooling too slow can be a waste of valuable computational resources and production time. Several advances in the formulation of cooling schedules for SA have been made in recent years [8, 19, 20]. Cooling schedules are discussed in the context of implementations in section 3.

2.2 Differences between Physical and Simulated Annealing

In SA, the investigator starts with a relatively small set of known or randomly sampled points in the domain. This differs significantly from physical annealing. For example, in the application of annealing to the production of glass lenses, the manufacturer can obtain a supply of raw silica which is equilibrated at some low temperature. In contrast, a random sample of points (microstates) for the optimization problem corresponds to <u>infinite</u> temperature in statistical mechanics. Consequently, trial values for T_0 are often estimated (or purposefully set [21]) too low. Such conditions can result in wild oscillations in temperature or non-equilibration in step **1c** of Algorithm 1.

This is sometimes termed “the warmup problem” [18].

A second major difference is that there is only one way to control temperature in SA: direct manipulation of the temperature parameter. In physical annealing, temperature can be controlled by applying a variety of energy sources or sinks.

In optimization problems, it is not always the case that $\#\mathcal{E} \ll \#\Omega$. A floating-point implementation will of course discretize the problem on a fine scale (about $O(10^{15})$ for double precision). Coarser scales can be obtained by modifying the cost function to “bin” energy values. Alternately, the move class can be tailored to select discrete values in the domain and thus “bin” energies in what may be nonuniform energy slices.

A final difference worth noting is that the number of microstates in any simulated ensemble will be very small in comparison to the theoretical number in real materials. This means that assumptions used in the derivation of annealing theory [6] may be invalid for statistically insignificant samples of the problem domain. Thus, it is important in any implementation of SA to (a) choose large enough N and (b) achieve a good distribution of values in step **1c** of Algorithm 1 before proceeding with step **2**.

3 Implementations

3.1 Monte-Carlo SA

The early days of computational SA [22] followed two decades of computational Monte-Carlo methods based on similar constructs [23, 24]. The capability of computing resources was minimal. Consequently, early algorithms were simplified and resembled Monte-Carlo trials. Calculations were often performed in batch runs as follows:

(Algorithm 2)

```
1. Initial set-up (manual):
    a. Select $N$ initial values $\{x\}$.
    b. Select or guess initial and final temperatures $T_0$ and $T_f$.
    c. Select temperature deflator $\alpha$ such that $T_f = \alpha^M T_0$ in $M$ steps.
2. Monte-Carlo runs. For each $x$:
    a. $T = T_0$ ; Write $x$, $F(x)$.
    b. Do $t = 1,\ M$
        • compute $T := \alpha T$.
        • Do $d = 1$, #neighbors($x$)
            – perturb the value of $x$ to some neighborhood value $x'$ with probability $\tau_{j \leftarrow i}$.
            – write $x$, $F(x)$.
    c. Store the output of each run in separate files.
3. Post-processing.
    a. Examine the output of all runs to determine what minima (if any) were found.
    b. Examine the output of each run to determine what distributions of energies were achieved.
    c. Utilize the output of b to formulate better estimates of $T_0$ and $T_f$ in future runs on same or similiar problems.
```

The capabilities of computing technology have advanced several orders of magnitude since the early days of SA. However, the basic form of this algorithm is still widely used. It has proven most effective in repetitive time-critical applications [25] and where background post-processing can be utilized to compute better initial values or static cooling schedules for future production runs [21].

3.2 Fixed-schedule SA

Algorithm 2 uses a fixed geometric cooling schedule in step **2b**. Geometric [11], linear [26], and other fixed schedules are pervasive in the SA literature. The best application of a fixed schedule is when an annealing schedule $T(t)$ derived from thermodynamic principles is known *a priori*. In absence of this, any other fixed schedule should be treated with extreme caution [18]. As in the annealing of real materials, the wrong schedule can lead to poor results.

3.3 Ensemble-based SA

To implement Algorithm 1 effectively, an ensemble size N should be selected which is appropriate for both the problem under study and the available computing resources [27, 7]. To estimate optimal N, the investigator can make pre-production runs of Algorithm 1 and store (among other things) the minimum energy seen up to time step t. The result will be an estimate of the "Very Best So Far Energy" distribution $VBSFE(t, N)$. Plotting $< VBSFE(t,N) >$ vs. N should reveal an $\hat{N}$ (or range of N values) which minimizes $< VBSFE(t,N) >$ [16].

Once N is determined, a set of initial values can be selected, either at random or by executing the move class $\frac{N}{K} - 1$ times on K known values. If the initial values are known a priori, then T_0 might also be known. If not, then T_0 is either unknown or infinite and in either case a finite value must be computed.

Intuitively, one would like to find $T_0 = T_{crit}$ which is "just hot enough" [28, 26]. The physical interpretation is that at high temperature the system will have a small relaxation time. As White [18] pointed out, this condition is

$$\frac{\sigma^2(T_0)}{T_0^2} \quad = \quad c(T_0) \ \ll 1 \tag{10}$$

The ratio $\frac{\sigma(T_0)}{T_0} \approx 0.01$ is a common trial value.

Equilibration at fixed T means that successive application of the move class to the ensemble with probability π causes little or no change in mean energy $\bar{E}(T)$. However, the implemented ensemble size will invariably be a small sample of the problem space. Thus, implementors should expect significant fluctuatons in $\sigma^2(T_0)$ unless great care has been taken to broadly sample the problem domain. A novice implementation may nonetheless encounter floating point exceptions. Using a cumulative distribution for $\sigma(T_0)$ over the first $d = \#neighbors$ time steps will avoid both anomalies.

Once T_0 has been established, the implementation can proceed with step 2 of Algorithm 1. If the cooling schedule is known, then the ensemble is cooled according to this schedule. If instead the cooling curve $\bar{E}(T)$ is known, the ensemble is equilibrated from T_0 to $\min(T)$ in increments $\Delta T(t) \approx \Delta^2 \bar{E}(t)$. In both cases, prior investigation(s) of the problem must be performed.

Initially, little is known about a problem with the possible exception of values discovered during the construction of a move class. These might help with the determination of T_0 and $\mathcal{B}(T)$ at high temperature, but not much else. The investigator now needs an adaptive method which can cool with optimal speed. To do so, equation 8 can be implemented either by adaptively constructing $\mathcal{B}_{ij}(T)$ to estimate $\zeta(T)$, or by using the "heat capacity" method [16] :

(Heat Capacity method for step 2,
Algorithm 1)

```
• $t = 0$
• $\bar{E}_0, \hat{\bar{E}}_0 = \bar{E}(T_0)$
• $\sigma_0 = \sigma(T_0)$
• while annealing do
    – $t := t + 1$
    – $\hat{\bar{E}}_t = \bar{E}_{t-1} - \nu \cdot \sigma_{t-1}$
    – $T_t = T_{t-1} + (\hat{\bar{E}}_t - \hat{\bar{E}}_{t-1}) \frac{T^2_{t-1}}{\sigma^2_{t-1}}$
    – move ensemble with probability $\pi$ and equilibrate at $T_t$
    – compute $\bar{E}_t, \sigma_t$
```

The algorithm owes its name to the appearance of $T^2/\sigma^2 = c$ in the temperature update step. This algorithm uses *target* mean energy values $\hat{\bar{E}}_t$ to implicitly estimate changes in T

in characteristic time $\zeta(T)$. The constant cooling speed ν should be set to a modest percentage value. If equation 9 has been observed in the construction of the move class, then $\nu =< \Delta\bar{E}(T) > / < \sigma(T) >$ will suffice. Equilibration at T_t can be estimated by repetitive application of the move class with probability π until $\bar{E}_t$ behaves more or less asymptotically, or by repetitive application for d steps, where $d \approx O(dimension(\Omega))$.

The question of when to stop annealing seems obvious enough to those with harsh time constraints. There are however penalties for annealing *too long*. The first penalty is related to the limited dynamics and size of the simulated ensemble. Some microstates will inevitably become trapped in "valleys" of less-optimal solutions and upwardly bias $\bar{E}$ and thus T. This will in turn slow progress towards better solutions. As progress slows, it is best to abandon states above a threshold (e.g., $\bar{E}$) and generate new states from the remaining set [16]. The second penalty is wasting time in the noise region of the system [13]. Once T is low enough that $|\Delta\bar{E}(T)| \approx T \approx$ "noise of the system" the simulation should be terminated.

3.4 Other Adaptive Schedules

Schedules similar in form to

$$\frac{dT}{dt} = \nu \frac{T}{\zeta(T)\sqrt{c(T)}}$$

(equation 8) have been discussed elsewhere in the literature. A method for cooling in constant energy slices was first developed at UC Berkeley [15] and later published in Numerical Recipies [10]. Kirkpatrick argued that dT/dt should somehow vary inversely to heat capacity [22] which has led others to ponder heuristics for incorporating heat capacity into adaptive schedules [21]. Ingber [12] has a freely available software package based on Gaussian probability distributions.

3.5 Parallel SA

We take the point of view that all algorithms are parallel until proven otherwise. The techniques of "porting a serial program to a parallel system" or "parallelizing an application program" are often mis-used as a starting point for the initial implementation of a parallel program. These methods are really only viable when the cost of re-writing a program is high in comparison with engineering a "port". Implementing a parallel version of an application directly from the underlying mathematics or system-level design is far more effective in producing efficient code than "porting" a program [29].

Fixed-schedule SA is *embarrassingly parallel* and hence no communication between parallel tasks is required. It is easily implemented on SIMD, MIMD, and Vector-Parallel[2] (VP) architectures. Adaptive SA methods may require communication of values for a few reduction calculations. However, these can be performed asynchronously and thus the computation should rarely wait on communication. In the case of $\bar{E}$ and σ, a global sum of energies can be requested to produce $\bar{E}$ on every processor, then a global sum of partial deviates can be requested to likewise produce σ.

SA algorithms will *vectorize* efficiently when individual vectors are used to store individual microstate data; e.g., vectors for E, previous $\bar{E}$, etc. Most modern RISC processors now support pipeline vector-like processing. Taking this approach is therefore beneficial to implementations running on both architectures.

Two levels of parallelism exist when implementing for MIMD architectures (including networks of workstations). At the gross system level there is the parallelism of multiple processors. At the processor level there is (or may be) vector or pipeline computation units. Thus the implementation is faced with the meta-optimization problem that seeks the optimal number of processors and states per processor which balances interprocessor communication time, memory per processor, processor speed, processor vector efficiency, and any I/O or device dependent operations. See Foster [29] for details on parallel program design.

---

[2] see [29] for a taxonomy

"Task Farming" is a parallel programming paradigm in which one processor serves as a master of task allocations and communications, while the other processors serve as computational slaves. Due to the high degree of parallelism in SA, this approach is a poor match. Rather than wait in linear time for reduction calculations, a strict SPMD implementation will wait at most in logarithmic time for global sums and then perform any remaining operations (such as a divide for $\bar{E}$) in parallel.

When creating data structures for use on symmetric multiprocessors (shared memory parallel architectures) the implementor should be careful to pre-allocate memory in a contiguious fashion, then provide some simple memory management scheme to use the preallocated memory as necessary. In particular, any linked-list structure should be preallocated at compile time as a finite-length array. The application can then build a linked list on top of the array with the added benefit of knowing when memory resources have been exhausted. This will avoid distributed memory fragmentations, a phenomena which can limit parallel speedup to $O(2)$ processors.

Acknowledgements

This paper would not have been possible without the patient advice and encouragement of Peter Salamon (SDSU), Paolo Sibani (Odense Universitet), and Reagan Moore (SDSC). Special thanks to Dyke Stiles (Utah State University) for organizing the 1997 PDPTA session on parallel stochastic optimization.

References

[1] David Boyd and John MacDowell, editors. *Advances in Ceramics: Commercial Glasses.* American Ceramic Society, Columbus, OH, 1986.

[2] Alexis G. Pincus and Thomas R. Holmes. *Annealing and strengthening in the glass industry.* Books for Industry, NY, NY, 1977.

[3] Minerals, Metals and Materials Society. *Developments in the annealing of sheet steels*, Warrendale, Pennsylvania, 1992.

[4] Risø National Laboratory. *Annealing processes– recovery, recrystallization, and grain growth*, Røskilde, Denmark, 1986.

[5] Donald A. McQuarrie. *Statistical Mechanics.* Harper and Row, NY, NY, 1975.

[6] E. Schroedinger. *Statistical Thermodynamics*, chapter 2. Dover.

[7] B. Andresen, K. H. Hoffmann, K. Mosegaard, J. Nulton, J. M. Pedersen, and P. Salamon. On lumped models for thermodynamic properties of simulated annealing problems. *J. Phys. France*, 49:1485–1492, 1988.

[8] J. Nulton and P. Salamon. Statistical mechanics of combinatorial optimization. *Physical Review A*, 37:1351, 1988.

[9] Otten and van Ginneken. Review of the annealing algorithm. *ACM Computing Reviews*, 31(6):296–298, June 1990.

[10] S. A. Teukolsky, W. T. Vetterling, and W. H. Press. *Numerical Recipes.* Cambridge University Press, http://cfatab.harvard.edu/nr/, 1993.

[11] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated annealing : theory and applications.* Kluwer Academic, 1989.

[12] Lester Ingber. Adaptive simulated annealing (ASA): Lessons learned. *Control and Cybernetics*, 1995. http://www.ingber.com/.

[13] R. Greening. *Simulated Annealing with Errors.* PhD dissertation, University of California, Los Angeles, Computer Science Department, 1995.

[14] Klaus Mosegaard and Peter D. Vestergaard. A simulated annealing approach to seismic model optimization with sparse prior information. *Geophysical Prospecting*, 39:599–611, 1991.

[15] F. Romeo, A. Sangiovanni Vincentelli, and C. Sechen. Simulated annealing research at Berkeley. In *IEEE Proceedings of the International Conference on Computer Design*, 1984.

[16] George Ruppeiner, Jacob Pedersen, and Peter Salamon. Ensemble approach to simulated annealing. *Journal de Physique*, 1(1):455–470, 1991.

[17] Stephen R. Wilson, Weili Cui, Jules W. Moskowitz, and Kevin E. Schmidt. Conformational analysis of flexible molecules: Location of the global minimum energy conformation by the simulated annealing method. *Tetrahedron Letters*, 29(35):4373–4376, 1988.

[18] S. R. White. Concepts of scale in simulated annealing. In *IEEE Proceedings of the International Conference on Computer Design*, page 646, 1984.

[19] E. Nørmark and K. Mosegaard. Residual statics estimation: Scaling temperature schedules using simulated annealing. *Geophysical Prospecting*, 41:565–578, 1993.

[20] P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures and Improvements for Simulated Annealing.* SIAM Monographs on Mathematical Modeling and Computation, in preparation (1997).

[21] D. S. Johnson, J. L. Bentley, L. A. McGeoch, and E. E. Rothberg. *Near-Opimal Solutions to Very Large Traveling Salesman Problems.* in preparation (1997).

[22] S. Kirkpatrick and C. D. Gelatt Jr. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[23] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

[24] C. J. Burke and M. Rosenblatt. *Annals of Math. Stats.*, 29:1112, 1958.

[25] F. H. Allisen Lee. *Parallel Simulated Annealing on a Large Message-Passing Multicomputer.* PhD dissertation, Utah State University, Electrical and Computer Engineering Department, 1995.

[26] G. Bilbro, R. Mann, T.K. Miller, W.E. Snyder, D.E. Van den Bout, and M. White. Optimizaton by mean field annealing. In *Advances in Neural Information Processing Systems.* Morgan Kaufmann, 1988.

[27] Karl Heinz Hoffmann, Paolo Sibani, Jacob M. Pedersen, and Peter Salamon. Optimal ensemble size for parallel implementations of simulated annealing. *Applied Mathematics Letters*, 3:53–56, 1990.

[28] A. Jagota, L. Sanchis, and R. Ganesan. Approximately Solving Maximum Clique using Neural Network and Related Heuristics. In D. S. Johnson and C. C. McGeoch, editors, *Proc. 1st DIMACS Implementation Challenge.* American Mathematical Society, 1993.

[29] Ian Foster. *Designing and Building Parallel Programs.* Addison-Wesley, 1995.