An ensemble based simulated annealing library for multiprocessors

Richard Frost *　Nicholas Priebe †　Peter Salamon ‡

5 October 1993

Abstract

We present a simulated annealing library for the Intel Paragon and other Unix-based systems. Our ensemble based, constant-speed approach is a true analog of annealing theory and compares favorably with other heuristic optimization techniques. Three types of parallelism are exploited for task distribution. The package includes diagnostic profiling to assist with the determination of annealing parameters. At the user's option some initializations can be performed adaptively. A sub-library for visual performance monitoring has been implemented on SGI workstations.

1 Introduction

Simulated annealing (SA) is an optimization procedure derived from the statistical mechanical theory for annealing real substances[19]. In recent years, SA has become a popular method for attacking problems with rough search spaces. It has a likely permanent niche in the arsenal of global optimization due to its computational simplicity.

There are several improvements to current SA implementations which are rarely taken advantage of due to the programming overhead required. Failure to implement these improvements can be extremely costly[30]. Among these, the poor initialization of walker distributions, cooling schedules, and stopping criteria are the most difficult barriers to researchers using simulated annealing. The analysis of numerical computational performance is also difficult due to long time scales and a moderate number of variable run-time parameters.

*San Diego Supercomputer Center (frost@sdsc.edu)
†University of California San Diego
‡San Diego State University

SA has developed over the years in two distinct fields: computational chemistry and numerical optimization. In the former, scientists simulate the cooling of liquids into *ground states* (glass) according to annealing theory. In the latter, computational scientists borrow from the same theory to "cool" the states of an optimization problem to a distribution of nearly optimal solutions.

It is important to note that SA models adopted by early optimization researchers[16] were highly constrained by compututational technology. Consequently their models were simple and resemble monte-carlo trials more than particle systems. When this work was re-discovered years later[13], many computer scientists did not wish to revisit annealing theory. Instead, the early models were taken as "first principle" and heuristics were often added in the hope of improving performance on specific problems[29]. As a result, the term *simulated annealing* now refers to a gamut of methods which may or may not resemble the physical theory.

2 Approach

We choose to represent the annealing theory as accurately as possible in our algorithm design. Thus, we use a statistically significant *ensemble* of domain elements to sample the cost function of an optimization problem[21]. The *walkers* that compose this ensemble explore the objective space in the same manner as the atoms or molecules of a liquid explore a potential surface. Further, they are subject to a *cooling schedule* which adaptively selects efficient transitions between global ensemble states—much like a glass maker tempers their ware.

Ensembles provide good statistics for adaptive temperature scheduling and a means of calculating metrics (e.g., Hamming distances) for studying the structure of configuration spaces. They also have the advantage of discovering ergodic breaks on their own[21] rather than requiring a (possibly false) partitioning of the configuration space. As a bonus, ensembles can yield a time-dependent thermodynamic portrait for calculating optimal cooling schedules and ensemble sizes of future production runs[9].

The question of find an optimal cooling schedule is still open although it is clear that adaptive schedules are better than non-adaptive ones. There are several around[3, 10, 4]. We believe in the optimality of constant speed[23]. This schedule has been independently proposed based on physical[5, 19, 23] and on biological reasoning[18]. It makes very significant (orders of magnitude) difference on some problems[7, 30].

Regardless of the selected cooling schedule, the initial collection of (possibly random) states must be equilibrated at some (high) temperature analog before cooling can begin. Consider that in the real world, we only cool real objects in some measurable state (e.g., a gram of silicon). To simulate annealing, we must therefore first *form* an object. This is done by collecting a statistically significant number of atoms (states of the cost domain) and subjecting them to the same heat bath until the system is within thermal noise of its infinite temperature equilibrium condition[31].

When equilibrium is acheived, the ensemble is then "cooled". Global statistics are sampled at each transition step and the system is forced to change at constant thermodynamic speed v. This is accomplished by determining the mean energy $\overline{E_i}$ at each transition step i and calculating a new *target* mean energy $\overline{E_{i+1}}^*$ which is v deviations from the current target $\overline{E_i}^*$. Either heat capacity or free energy relationships can then be used to establish what change in temperature is theoretically necessary to acheive $\overline{E_{i+1}}^*$. Further details can be found in [21] and other references listed in the bibliography.

3 Software

We have implemented our approach in an ANSI-C library called EBSA. To use our software, researchers must write their own definition of an objective function, domain, and move class. They may then link to our library to perform the optimazation task(s). We feel that this approach gives users freedom to define their own data structures while protecting them from many pitfalls in SA implementation. A script called *ebsagen* is provided to help users generate sample main programs. Documentation and a library of sample objective functions, move classes, etc. is also included. EBSA is available free via anonymous ftp.

Ensemble based SA lends itself naturally to SIMD, SPMD, and MIMD programming models. Our software runs on Unix workstations, MIMD architectures such as the Intel Paragon, and shared memory parallel platforms such as the Cray YMP. Algorithm parallelization is only limited by the users objective function. That is, the user's objective function will parallelize only to the extent of parallelism imbedded in their implementation.

A graphical display for the interactive analysis of run-time performance is one of EBSA's primary features. Ten different diagnostic charts are provided to monitor the progress of the optimization process and the effectiveness of the user's move class. The user can also supply additional graphic metrics, permitting EBSA to visually depict selected states of the domain.

4 Work In Progress

We have identified six key research sub-topics within our work. Some are currently under study. Work is scheduled to begin on the remainder in 1994.

4.1 The cooling schedule

Some development is needed for efficient mappings of continuous variable problems with free-energy methods. Otherwise, we hope that the free distribution of our software will allow constant thermo-speed methods to be tried on a variety of problems. The subsequent findings are likely to help in proving optimality by the techniques of Hoffmann & Salamon [9].

4.2 Move Class

Most problems offer several alternative techniques of perturbing (mutating) configurations to give other candidate configurations. The move class serves to define a graph structure on the set of configurations. Optimal blend of move classes is an open problem which has led to some heuristics such as trying to maintain an acceptance rate around 50% [15, 2]. Other global optimization schemes (such as genetic algorithms) invest a great deal of energy in designing and implementing efficient crossover moves. Tools for comparing the effectiveness of different move classes (including crossovers) in generating promising candidate configurations will be developed.

4.3 Scaling properties

The problem of time complexity of simulated annealing implementations is open and could be studied directly with the data gathered by existing software diagnostics. EBSA automatically generates thermodynamic portraits and thus serves to examine the scaling behavior of constant speed schedules. Such scaling properties are of course dependent on the move class and the cooling schedule. It is expected that knowledge of the scaling properties of a problem would also enable the package to implement adaptively optimal ensemble sizes[9].

4.4 The structure of complex configuration spaces

The detailed structure of configuration spaces, albeit dependent on the chosen move class, can provide crucial data for selecting methods for attacking problems. Many hints indicate that the structure of some of these configuration spaces shows a type of dynamic self similarity[27]. Given the structure of a local configuration pocket, optimal control techniques can yield optimal cooling schedules for annealing within such pockets. Assuming self similarity is present, the schedules obtained from optimal control can be extended to optimal schedules for the entire annealing run. The study of the structure of configuration spaces will also explore and attempt to prove the "Deep wells have large rims" ansatz. This is crucial for proving the optimality of algorithms which allocate search effort to components according to the volume of available configuration space.

4.5 Objective functions with external noise

Many real problems of practical interest intrinsically include noise in the objective function. Examples include seismic deconvolution[14] and more generally pattern recognition problems based on the processing of signals from a noisy channel. For such problems, the function which is really to be minimized differs from the available function evaluated in the algorithm. Thus the values of the objective are accurate only to a certain tolerance. We plan to provide a runtime implementation for stopping the annealing process at the "noise" temperature.

4.6 Information theoretic analysis of global optimization algorithms

The larger goal of our work is to lay more of the foundation for proving optimality of adaptive ensemble based simulated annealing for problems in which only two procedures are available: move generation and energy evaluation. The expected analysis will encompass crossover moves and thus genetic algorithms. Theorems regarding optimal strategies of sampling and acceptance are expected.

5 Acknowledgements

This work was funded in part by NSF grant ASC–8902825. Special thanks to Intel Supercomputer Systems Division for their sponsorship of ISUG–93.

References

[1] 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, 1988.

[2] K. Binder. *Monte Carlo Methods in Statistical Physics*. Springer Verlag, 1979.

[3] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *PAMI*, 6:721–741, 1984.

[4] Bruce Hajek. Cooling schedules for optimal annealing. *Math. Oper. R.*, 13:311–329, 1988.

[5] John Ro Harland and Peter Salamon. Simulated annealing: A review of the thermodynamic approach. *N. Phys. B (Proc. Suppl.)*, 5A:109–115, 1988.

[6] K. H. Hoffmann, D. Wuertz, C. de Groot, and M. Hanf. Concepts in optimizing simulated annealing schedules: An adaptive approach for parallel and vector machines. In M. Grauer and D.B. Pressmar, editors, *Parallel and Distributed Optimization*. Springer-Verlag, Heidelberg, 1991.

[7] Karl Heinz Hoffmann and Peter Salamon. The optimal simulated annealing schedule for a simple model. *Journal of Physics A*, 23:3511, 1990.

[8] Karl Heinz Hoffmann and Peter Salamon. Simulated annealing for single minimum optimization problems. *International Journal of Computer Mathematics*, 39:193–204, 1991.

[9] Karl Heinz Hoffmann, Paolo Sibani, Jacob M. Pedersen, and Peter Salamon. Optimal ensemble size for parallel implementations of simulated annealing. *Appl. Math. Lett.*, 3:53–56, 1990.

[10] M. D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli. An efficient general cooling schedule for simulated annealing. In *ICCAD 1986*, page 381. IEEE, 1986.

[11] E.T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620, 1957.

[12] E.T. Jaynes. Information theory and statistical mechanics II. *Phys. Rev.*, 108:171, 1957.

[13] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. *Science*, 220:671, 1983.

[14] Z. Koren, K. Mosegaard, E. Landa, P. Thore, and A. Tarantola. Seismic background velocity estimation and model error analysis by simulated annealing. *Journal of Geophysical Research*, 96(B12):20289–20299, 1991.

[15] J. Lam and J. M. Delosme. An adaptive annealing schedule. Technical Report 8608, Department of Electrical Engineering, Yale University, 1989.

[16] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. *J. Chem. Phys.*, 21:1807, 1953.

[17] I. Morgenstern and D. Wuertz. Simulated annealing for spin-glass-like optimization problems. *Z. Phys. B*, 67:397, 1987.

[18] H. Muehlenbein and D. Schleirkamp-Voosen. Predictive models for the breeder genetic algorithm: I. continuous function optimization. *Evolutionary Computation*, 1:25–49.

[19] James D. Nulton and Peter Salamon. Statistical mechanics of combinitorial optimization. *Phys. Rev. A*, 37:1351–1356, 1988.

[20] George Ruppeiner. Implementation of an adaptive, constant thermodynamic speed simulated annealing schedule. *N. Phys. B*, 5A:116–121, 1988.

[21] George Ruppeiner, Jacob M. Pedersen, and Peter Salamon. Ensemble approach to simulated annealing. *J. Phys. I*, 1:455–470, 1991.

[22] Peter Salamon, Karl Heinz Hoffmann, John Ro Harland, and James D. Nulton. An information theoretic bound on the performance of simulated annealing algorithms. Technical Report IRC.89.2, Interdisciplinary Research Center, San Diego State University, 1989.

[23] Peter Salamon, James D. Nulton, John R. Harland, Jacob Pedersen, George Ruppeiner, and Luby Liao. Simulated annealing with constant thermodynamic speed. *Comp. Phys. Comm.*, 49:423–428, 1988.

[24] P. Sibani and K. H. Hoffmann. Random walks on cayley trees. *Europhysics Letters*, 4:967–972, 1987.

[25] P. Sibani and K. H. Hoffmann. Relaxation in complex systems: Local minima and their exponents. *Europhysics Letters*, 16:423–428, 1991.

[26] P. Sibani, J. M. Pedersen, K. H. Hoffmann, and P. Salamon. Monte carlo dynamics of optimization problems, a scaling description. *Phys. Rev. A*, 42:7080–7086, 1990.

[27] P. Sibani, J. C. Schön, P. Salamon, and J. O. Andersson. Emergent hierarchical structures in complex system dynamics. *Europhysics Letters*, 22:479–485, 1993.

[28] P.F. Stadler and W. Scnabl. The landscape of the tsp. *Phys. Lett. A*, 161:337, 1992.

[29] P.J.M. van Laarhoven and E.H.L. Aarts. *Simulated annealing: theory and applications.* Kluwer Academic, 1987.

[30] P. D. Vestergaard and K. Mosegaard. Inversion of post-stack seismic data using simulated annealing. *Geophysical Prospectin*, 39:613–624, 1991.

[31] Steven R. White. Concepts of scale in simulated annealing. In *ICCAD '84 Proc.*, pages 646–651. IEEE, 1984.

[32] Jea-Yel Yi, Jerry Bernholc, and Peter Salamon. Simulated annealing strategies for molecular dynamics. *Comp. Phys. Comm.*, 66:177–180, 1991.

[33] Theo Zimmermann and Peter Salamon. The demon algorithm. *Intern. J. Computer Math.*, 42:21–31, 1992.