[17] S.-Y. Lee and J.K. Aggarwal. A mapping strategy for parallel processing. *IEEE Trans. on Computers*, C-36(4):433–442, April 1987.

[18] P. Molitor. Layer assignment by simulated annealing. *Microprocessing and Microprogramming*, 16:345–350, 1985.

[19] C.D. Polychronopoulos and U. Banerjee. Processor allocation for horizontal and vertical parallelism and related speedup bounds. *IEEE Trans. on Computers*, C-36(4):410–420, April 1987.

[20] R. Rammal, G. Toulouse, and M.A. Virasoro. Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765–788, July 1986.

[21] F. Romeo, A. Vincentelli, and C. Sechen. Research on simulated annealing at berkeley. In *IEEE Int'l. Conf. Computer Design*, pages 652–657, 1984.

[22] P. Sadayappan and F. Ercal. Nearest-neighbor mapping of finite element graphs onto processor meshes. *IEEE Trans. on Computers*, C-36(12), December 1987.

[23] S.A. Solla, G.B. Sorkin, and S.R. White. Configuration space analysis for optimization problems. In Bienenstock et al., editors, *Disordered Systems and Biological Organization*, pages 283–293. Springer-Verlag, 1986.

[24] G.B. Sorkin. Simulated annealing on fractals: Theoretical analysis and relevance for combinatorial optimization. In W.J. Dally, editor, *Advanced Research in VLSI*, pages 331–351. MIT Press, 1990.

[25] M. Stefik and D.G. Bobrow. Object-oriented programming: Themes and variations. *AI Magazine*, 6(4):40–62, January 1986.

References

[1] D.H. Ackley. *Stochastic Iterated Genetic Hillclimbing.* PhD thesis, Carnegie Mellon University, March 1987. CMU-CS-87-107.

[2] G.A. Agha. Actors: A model of concurrent computation in distributed systems. Technical report, M.I.T. Artificial Intelligence Laboratory, Tech. Rep. 844, June 1985.

[3] J.R. Anderson and C. Peterson. Applicability of mean field theory neural network methods to the graph partitioning problem. Technical report, MCC Tech. Rep. ACA-ST-064-88, February 1988.

[4] J. Banerjee et al. Data model issues for object-oriented applications. *IEEE Trans. on Office Info. Sys.*, 5(1):3–26, January 1987.

[5] M.J. Berger and S.H. Bokhari. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Trans. on Computers*, C-36(5):570–580, May 1987.

[6] F. Berman et al. Prep-p: A mapping preprocessor for chip computers. In *International Conference on Parallel Processing*, pages 731–733, 1985.

[7] L. Bic and R. Hartmann. The active graph database model. In *Proc. 2nd Int'l Conference on Databases, ICOD-2*, 1983.

[8] L. Bic and C. Lee. A data-driven model for a subset of logic programming. *IEEE Trans. on Prog. Lang. and Sys.*, 9(4):618–645, October 1987.

[9] S.H. Bokhari. On the mapping problem. *IEEE Trans. on Computers*, C-30(3):550–557, March 1981.

[10] A.L. Davis and R.M. Keller. Data flow program graphs. *Computer*, pages 26–41, February 1982.

[11] K. Fukunaga, S. Yamada, and T. Kasai. Assignment of job modules onto array processors. *IEEE Trans. on Computers*, C-36(7):888–891, July 1987.

[12] J.L. Gaudiot, J.I. Pi, and M.L. Campbell. Program graph allocation in distributed multiprocessors. *Parallel Computing*, 7:227–247, 1988.

[13] S. Goto. An efficient algorithm for the two-dimensional placement problem in electrical circuit layout. *IEEE Trans. Circuits and Systems*, CAS-28(1):2–18, January 1981.

[14] C.E. Hewitt. Viewing control structures as patterns of passing messages. *J. of Artificial Intelligence*, 8(3):323–364, June 1977.

[15] Y.-C. Hong, T.H. Payne, and L.B.O. Ferguson. Graph allocation in static dataflow systems. In *International Symposium on Computer Architecture*, pages 55–64, 1986.

[16] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, May 13 1983.

distance between two configurations implies that their energies will generally differ by less. The fact that quenched solutions with a higher energy can't find a closer annealed solution implies that the configuration space itself has a regular shape, that "valleys" with the same shape have multiple occurances such that a quenching that lands with energy E in one valley may be distant from an annealing with energy $E - \epsilon$ in another valley but has a closer isomorphic permutation in that same valley.

Such an interpretation is consistent with the argument given by Solla et al. in support of the correlation between move distance and ultrametricity. They speculate that higher ultrametricity is observed when there is a correlation between move distance and the *barrier energy*. The barrier energy between two configurations is the minimax energy over all paths between them. When the barrier energy is used as the distance metric between two configurations, the space is ultrametric. This follows directly from the definition of barrier energy. When a correlation exists between move distance and barrier energy, then ultrametricity is observed when move distance is used as the metric. If many similarly shaped valleys exist, then it is likely that such a correlation exists, i.e., that low-lying minima are evenly and ultrametrically distributed in the configuration space. However, it is an open question whether these configuration spaces have 'self-similar' or 'fractal-like' valleys, as described in [24], such that similarly-shaped valleys occur with a recursively decreasing size. Such a property can increase the effectiveness of annealing since the size of a valley with a local minima that should be avoided decreases as the annealing schedule progresses to lower and lower temperatures.

The observations discussed so far lead us to conjecture that regularity in the network and problem graphs – and any optimization problem that contains similar regularity – produces a configuration space that reflects the same structure in which low-lying minima are ultrametrically distibuted. What impact, however, does this have on the relative performance of annealing versus quenching? Higher regularity produced more closely clustered energy values for both annealing and quenching. This implies that when higher regularity exists, a shorter annealing schedule can be used with a higher probability that the resulting maps will not be far apart in energy.

Graph Type	Quenching	Annealing
perfect	0.205	0.473
1-displaced	0.148	0.402
2-displaced	0.090	0.253
4-displaced	0.041	0.138

Table 2: Ultrametric Correlation Coefficients

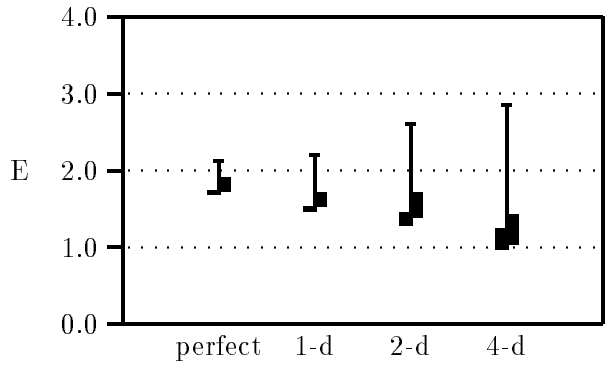

Figure 10: Range of Absolute Energy Values

5 Summary and Conclusions

In this paper, we have endeavored to understand the effect of structure on the relative performance of annealing and quenching. This was motivated by our research in parallel processing where problem graphs of varying regularity must be mapped onto very regular network graphs under the constraints of (1) minimized communication distance, and (2) even distribution over the network. To capture these two requirements, we defined the distance-variance energy metric. The conclusions given here are based on this metric and may not be general for all possible metrics. All metrics may not exhibit the sensitivity to structure as displayed by the distance-variance metric. It is possible, however, that any metric for the mapping problem that is based on uniformity and the minimization of mapped edge distance will exhibit the same properties.

These experiments have shown that higher regularity in the problem graph produces (1) a narrower but higher range of final energy values, (2) a correlation between move distance and energy, and (3) a higher ultrametric correlation. What can we conclude from this? It seems fairly safe to assume that higher regularity produces a landscape that is "flatter" with a narrower distribution of local minima. The relationship, however, between the move distance and energy difference is less obvious.

Why does higher graph regularity produce a definite distance-energy correlation? One argument is that finding the closest map permutation ensures that distances will be reduced when dealing with a regular graph since such a graph has many isomorphic permutations. Since all edges and vertices contribute to the energy metric, a smaller move

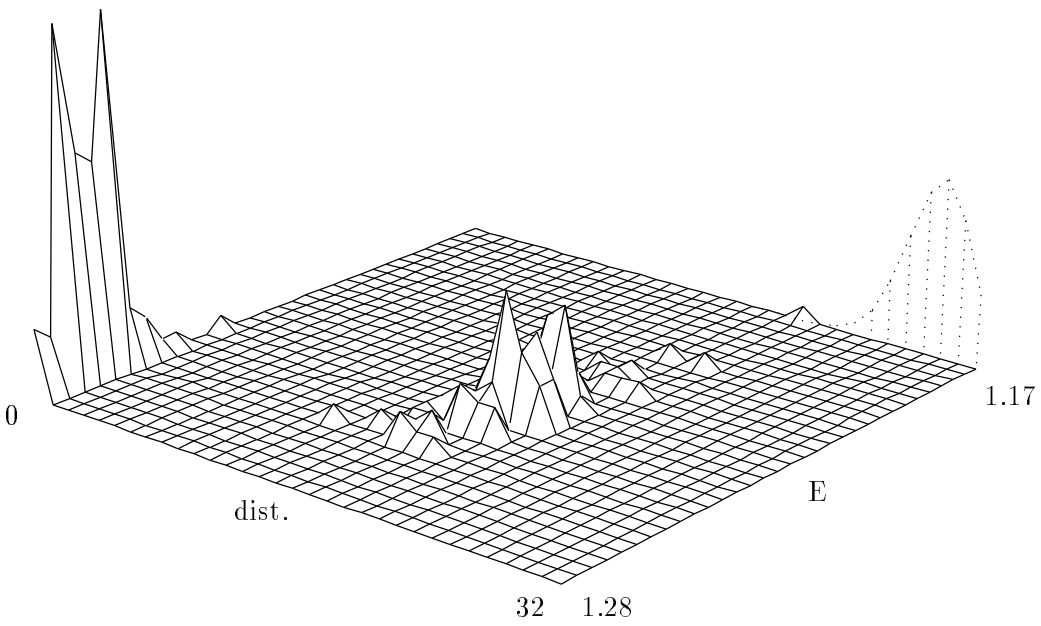

Figure 8: Distance-Energy Surface for Torus with Two Displacements

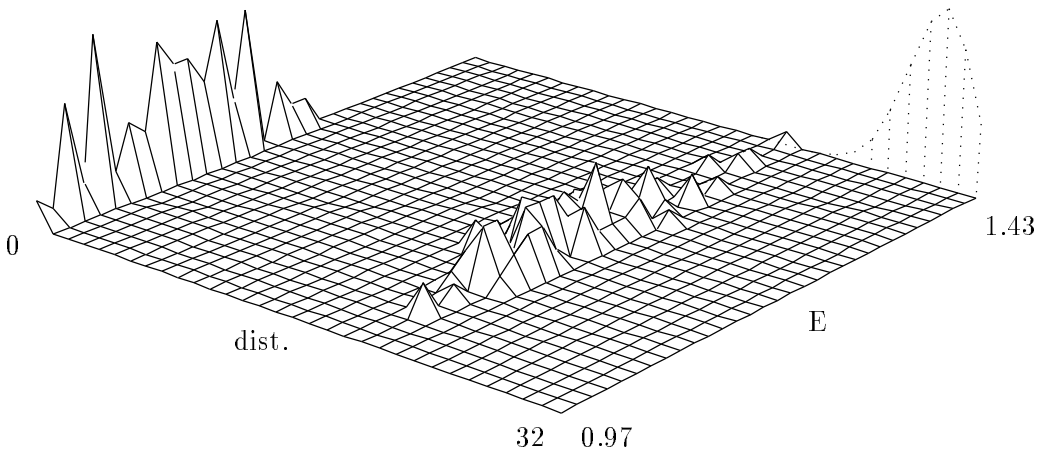

Figure 9: Distance-Energy Surface for Torus with Four Displacements

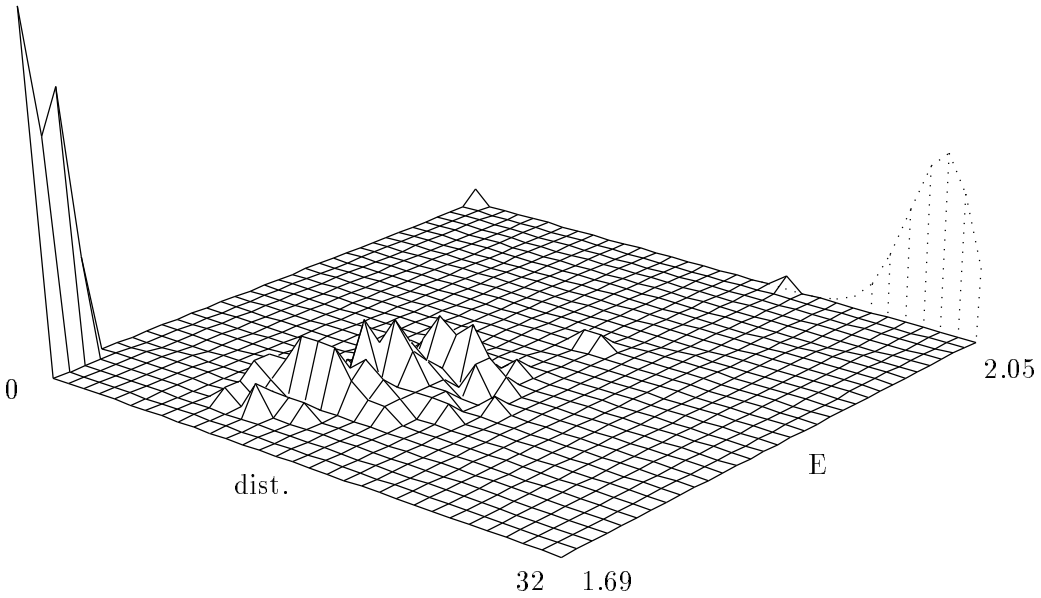

Figure 6: Distance-Energy Surface for Perfect Torus

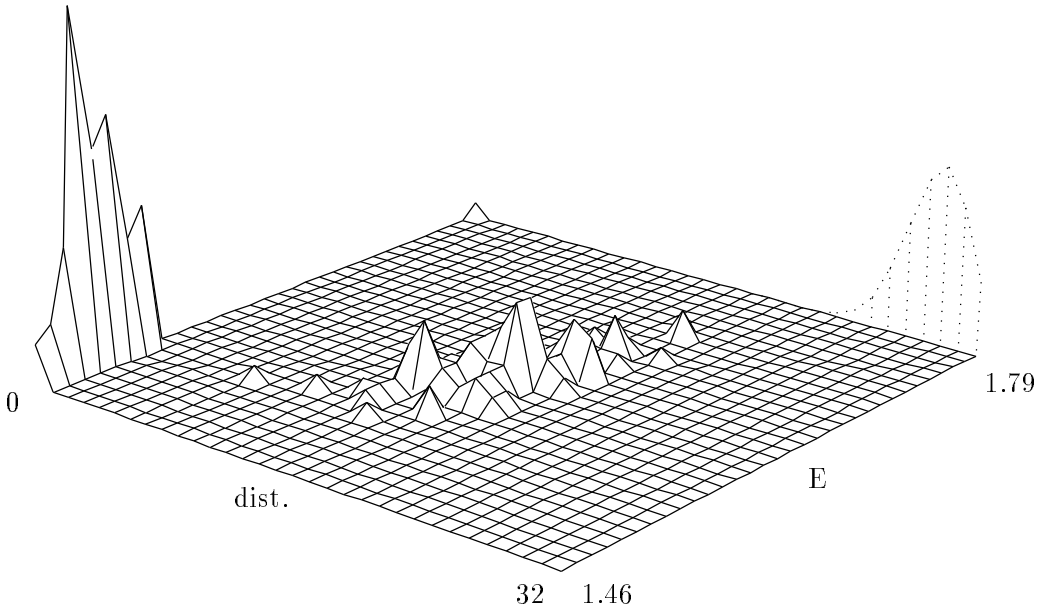

Figure 7: Distance-Energy Surface for Torus with One Displacement

distribution of local minima that trap annealing and quenching.

4 Behavior under Edge Displacement

How sensitive is the current energy function to changes in the regularity of a graph? It was somewhat surprising that there was no significant difference between the random and geometric graphs in the observed ultrametricity. Likewise, both graph types had very similar final energy ranges. By essentially confining each edge in the geometric graph to a "neighborhood", its similarity with the network graph was increased. This increase in similarity, however, did not translate into observable ultrametricity.

Hence, we investigated the sensitivity of the distance-variance metric by mapping a 4×8 standard torus problem graph (each vertex with four nearest-neighbor edges) onto the same 3×3 torus network graph while displacing 1, 2, or 4 edges in the problem graph. The displaced edges in the problem graph were chosen at random yet the displacements simply moved one endpoint of an edge one vertex further away.

The same data was collected for torus problem graphs with 0, 1, 2, and 4 displaced edges; 100 annealings and 100 quenchings were done using the same annealing parameters as stated in the previous section. Figures 6 to 9 plot the energy-distance differences between the annealed and quenched solutions in the same manner as Figures 1 to 4. These surfaces show that the correlation between distance and energy degrades quickly as the regularity of the problem graph is compromised. This is reflected in the observed ultrametricity coefficients shown in Table 2. By the time that four edges have been displaced, the correlation and ultrametricity have gone from that of a perfect torus to that of an essentially random graph. The presence of four slightly out-of-place edges (out of 64) made the configuration space much less amenable to simulated annealing.

This progressive effect is also seen in the final energy range as shown in Figure 10. The range for the perfect torus is very similar to that of the torus in the previous section. Displacing four edges, however, causes the initial energy to rise and the final energy range to fall. (Note that these bars exclude solutions trapped in high local minima.) The size of the range is small for the perfect torus and broader with four displaced edges; much like that of the random and geometric graphs.

Graph Type	Quenching	Annealing
random	0.028	0.036
geometric	0.033	0.035
torus	0.238	0.401
n-cube	0.167	0.222

Table 1: Ultrametric Correlation Coefficients

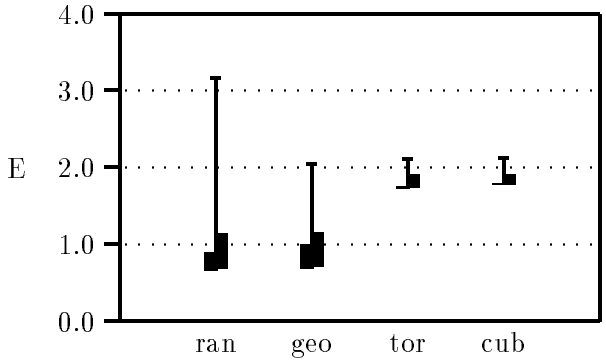

Figure 5: Range of Absolute Energy Values

results; there is a significant range of annealed energies and an even broader range of energies for the quenched solutions. The additional "structure" in the geometric graph did not significantly change the distribution of solutions found. All of the quenched solutions are in a narrow range of distances from their closest annealed solutions. There is no apparent correlation between the distance between solutions and their difference in energy. In fact, the distribution of quenched solutions seems strongly influenced by the probability of finding a close configuration within the general distribution of configurations.

The torus and n-cube graphs, however, tell a different story. The annealed solutions have a tightly clustered energy range. (For the sake of typesetting, the peaks in these surfaces have been truncated.) The quenched solutions show a definite correlation between distance and energy aside from a few solutions trapped in high local minima. The ultrametric correlation coefficients were computed as discussed in the previous section and are shown in Table 1. This clearly supports the notion that simulated annealing performs better in a configuration space that exhibits ultrametricity which is, in turn, related to the correlation between distance and energy difference. What the graph types tell us is that this correlation is dependent on the regular and self-similar structure present in the problem.

The differences discussed so far, however, are *relative* differences. Figure 5 shows the absolute energy range for quenching and annealing for the four different graph types including the initial map energy. This shows that while the random and geometric graphs start at a equal or much higher energy, they finish in an energy range that is broader and much lower than the torus and n-cube. (Note that these bars exclude solutions trapped in high local minima.) What this tells us is that graph structure affects the size and

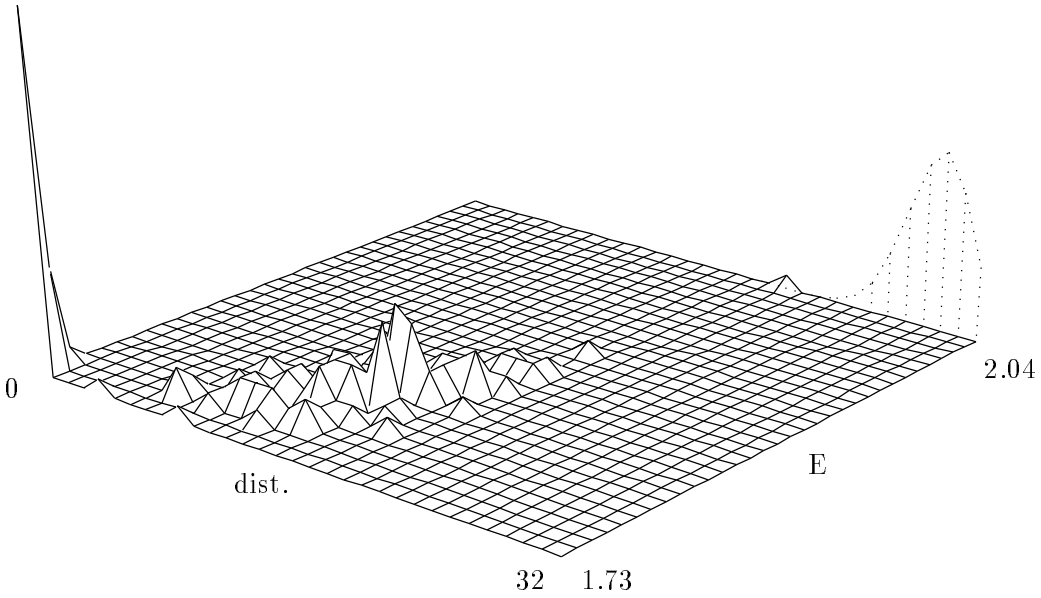

Figure 3: Distance-Energy Surface for Torus Problem Graph

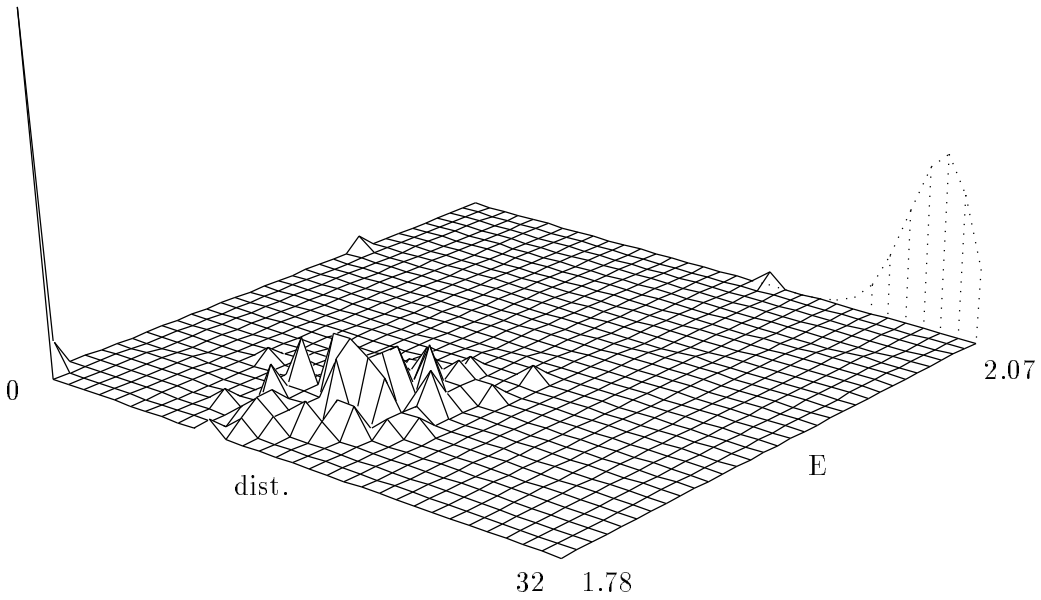

Figure 4: Distance-Energy Surface for n-Cube Problem Graph

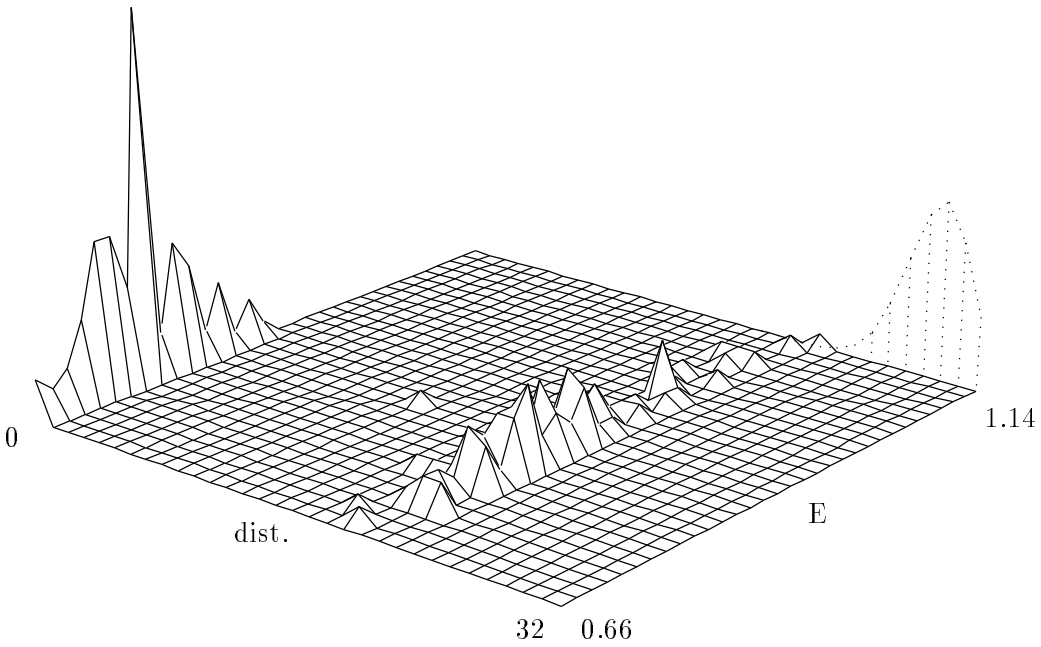

Figure 1: Distance-Energy Surface for Random Problem Graph

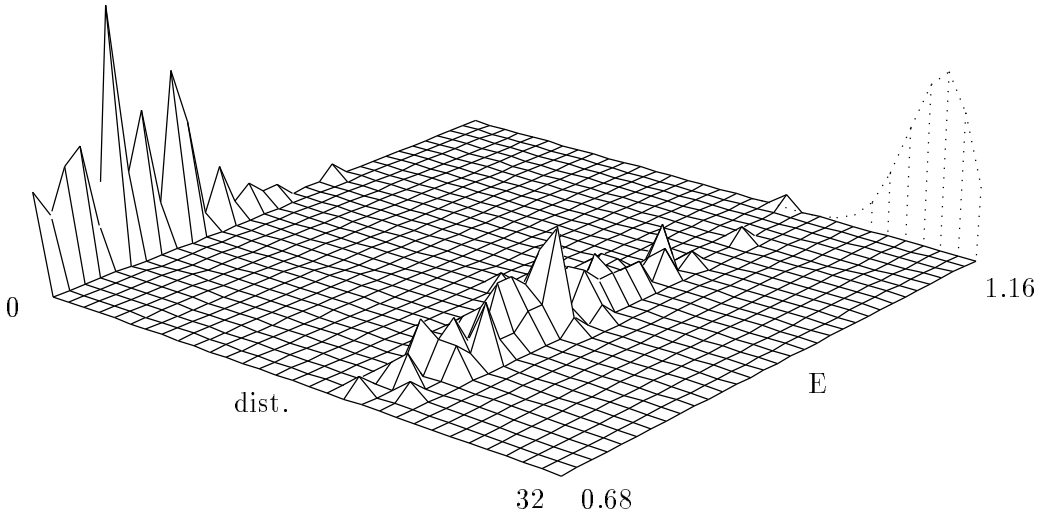

Figure 2: Distance-Energy Surface for Geometric Problem Graph

The torus and n-cube are very regular in their structure but are similar and dissimilar, respectively, to the network graph.

These graphs are defined in the following way. All problem graphs have 32 vertices and an average vertex degree of 5. This ensures that any difference in numerical energy results from differences in graph structure, i.e, how well the graph can be mapped, rather than just the number of vertices and edges involved. A random graph is generated by defining n vertices and connecting any two with probability p which results in an average degree of $(n-1)p$ [3]. Given the desired degree of 5, it is easy to compute the required p. A geometric graph is generated by randomly distributing n vertices over the unit square and connecting any two vertices that are contained within a square with side $0 < k < 1$ where this smaller square can wrap-around to the other sides of the unit square. This definition is intended to produce a topology similar to a torus. (If wrap-around was not allowed, the generated graph would be more similar to the layout of a printed circuit board as defined in [3].) Here the average degree is $4nk^2$. Given the desired degree of 5, it is easy to compute the required k. The problem torus will be a 4×8 grid but with extra edges added in a symmetric manner such that each vertex has a degree of 5. Finally, the n-cube with 32 vertices naturally has a degree of 5. Finally, the network graph will be a 3×3 torus; each vertex will have a degree of 4 in the typical grid fashion. This network graph may seem small but it is not a degenerate torus and it was deemed necessary to hold down the cost of checking permutations. All of these graphs may seem small but each configuration space has a size of 9^{32}.

For each problem graph, we will do 100 annealings and 100 quenchings onto the network graph. In all cases, the same initial mapping will be used but with a different seed for the random number generator. In all cases, an even 0.5/0.5 weighting will be used in the distance-variance energy function. The *temperature* control parameter has an initial value of 100 ($\approx 50\times$ the initial energy value) and an exponential decay factor of 0.9. At each temperature, thermal equilibrium will be determined as in [16]: for a problem of size n, equilibrium is reached after an changes have been accepted or bn changes have been attempted (accepted or not), whichever comes first. The user is free to choose a and b under the constraint that $0 < a \leq b$. For quenching, $a/b = 1/2$. For annealing, $a/b = 128/256$. The algorithm terminates when the configurations at the last three consecutive temperatures show no improvement, i.e., the configuration is "frozen" [21].

For each quenched solution within each graph type, the map permutation with the minimum distance to any annealing solution is found. We then plot the distance versus the energy as a histogram surface as shown in Figures 1 to 4. All annealed solutions are set at distance zero and the distance axis shows how far away the closest quenched solution permutations are. Since the problem graphs all have 32 vertices, all distances are in the range $\langle 0, 32 \rangle$. The energy axis, however, is scaled to the minimum and maximum values of all annealings and quenchings and then quantized into 32 bins. The vertical axis then shows the number of solutions in a particular bin. The vertical scale is arbitrary but uniform across all graphs. This graphical display shows at a glance the distribution of the annealed and quenched solutions relative to one another. For comparison, the general distance distribution of all possible configurations from any one configuration is shown as a dotted curve.

Figures 1 to 4 show the following. The random and geometric graphs have very similar

An ultrametric space satisfies instead

$$iii') \quad d(x,z) \leq \max[\, d(x,y),\, d(y,z)\,]$$

An example of an ultrametric space is a complete tree; any three leaves x, y and z satisfy i, ii and iii' where the distance between any two leaves is the length of the shortest path between them.

How well a space satisfies the ultrametric property can be tested using *triangle statistics*. In an ultrametric space, any three states form a triangle and the two longer sides are expected to be equal. This can be evaluated by the subtracted probability function [3, 20, 23]

$$\tilde{C}(d_1, d_2) = \tilde{P}(d_1, d_2) - \tilde{P}(d_1)\tilde{P}(d_2)$$

where $\tilde{P}(d_1, d_2)$ is the probability that d_1 and d_2 are the two longer sides of a triangle and $\tilde{P}(d)$ is the probability that d is the longest side of a triangle.

In terms of the mapping problem, the distance between any two maps can be defined as the number of elements where the maps differ. Since some pairs of maps may actually be isomorphic, we must strengthen the definition of distance to be the minimum distance between any isomorphic permutations of the maps. Hence, we can compute $\tilde{C}(d_1, d_2)$ by generating a number of maps and collecting the probability of distances occurring in all possible triangles. This can be plotted as a surface that is symmetric about the diagonal. A truly ultrametric space exhibits a high correlation along the diagonal where $d_1 = d_2$ and an anti-correlation off the diagonal. Such a visual correlation can be quantified in a correlation coefficient given by Solla et al.:

$$c = \frac{\langle (d_1 - \langle d_1 \rangle)(d_2 - \langle d_2 \rangle) \rangle}{\sigma_1 \sigma_2}$$

and

$$\sigma_i = \langle (d_i - \langle d_i \rangle)^2 \rangle^{1/2}$$

where $\langle ... \rangle$ indicates average with respect to the joint probability distribution $\tilde{P}(d_1, d_2)$.

The interpretation of $\tilde{C}(d_1, d_2)$ very interesting. In a sense, it reduces a n-dimensional space down to three which can be easily plotted as a surface. It preserves the notion of space and distance by the use of triangles; if a number of solutions are clustered, there is an even greater number of triangles contributing to the distances that characterize the cluster. If there are multiple clusters, however, the distances between clusters also contributes to the probabilities. Indeed, $\tilde{C}(d_1, d_2)$ can indicate clusters and the distance between clusters but not their number or overall distribution in the solution space.

3 Four Graph Types

Having introduced the mapping problem, the energy metric of interest to us and ultrametricity, we now investigate the effect of problem structure on the behavior of simulated annealing and quenching by mapping members of four different graph types, (1) random, (2) geometric, (3) torus, and (4) n-cube, onto a torus. These graph types were chosen to represent a range of "structure". The random graph can be said to have no structure. The geometric graph, as defined below, has more structure with similarities to a torus.

is mapped only half-way across the network:

$$QD_{mode} = \frac{|E_p| \cdot diam}{2}$$

The minimum LV occurs when there is a perfect distribution. The maximum LV occurs when all problem graph edges are mapped to one PE. Hence, we have

$$\begin{aligned} LV_{min} &= 0 \\ LV_{max} &= \frac{(|V_n|-1){d_{avg}}^2 + (2 \cdot |E_p| - d_{avg})^2}{|V_n|} = 4 \cdot |E_p|^2 \cdot \left(\frac{|V_n|-1}{|V_n|^2} \right) \end{aligned}$$

where

$$d_{avg} = \frac{2 \cdot |E_p|}{|V_n|}$$

is the average logical degree for any PE. The mode of the distribution of the logical variance, however, does not fall in the middle of this range; it is extremely skewed towards LV_{min}. LV_{mode} occurs when the logical degree varies from the average by one half. Hence,

$$LV_{mode} = \frac{\Sigma(\frac{3 \cdot d_{avg}}{2} - d_{avg})^2}{|V_n|} = \frac{|E_p|^2}{4 \cdot |V_n|^2}$$

Using these mode values, we can scale the metrics and weight them:

$$E = W_{qd} \frac{QD}{QD_{mode}} + W_{lv} \frac{LV}{LV_{mode}}$$

where W_{qd} and W_{lv} are the weights with the constraints that

$$0 \leq W_{qd}, W_{lv} \leq 1 \qquad \text{and} \qquad W_{qd} + W_{lv} = 1$$

This weighting problem can be avoided in the special case where $|V_p| \leq |V_n|$. LV can be simplified by requiring that each problem vertex is mapped onto a different PE, i.e., that m be one-to-one, as in the approach described in [9]. Pairwise exchanges of problem vertices during the optimization process do not alter the distribution of the problem (the LV metric) and hence the energy function can use only the QD metric to derive a solution.

2.3 Ultrametricity

When applied to a mapping problem, the energy function defined in the previous section will determine a configuration space as discussed in Section 1. We will use ultrametricity to study the structure of these spaces. Ultrametricity is thoroughly reviewed in [20] and has been applied to the mapping problem [3] but not to make a comparison between annealing and quenching concerning different graph types.

Ultrametricity essentially tests for the hierarchical ordering of states by replacing the triangle inequality usually associated with metric spaces. Typically, a metric space with a distance function d satisfies

$$\begin{aligned} i) \quad d(x,y) &= 0 \iff x = y \\ ii) \quad d(x,y) &= d(y,x) \\ iii) \quad d(x,z) &\leq d(x,y) + d(y,z) \end{aligned}$$

2.2 The Distance-Variance Energy Function

The mapping problem can be stated as follows: given a network graph $G_n = \langle V_n, E_n \rangle$ and a problem graph $G_p = \langle V_p, E_p \rangle$, find a mapping $m : V_p \mapsto V_n$ that maximizes or minimizes some metric E. When mapping a problem graph onto a network, the work should be evenly distributed across the entire network while minimizing communication overhead. These are competing goals.

To minimize the communication overhead, adjacent vertices in the problem graph must be mapped as close together in the network as possible. Hence, the sum of the network distances between any two adjacent problem vertices,

$$QD = \sum_{\langle v_1, v_2 \rangle \in E_p} dist(\ m(v_1),\ m(v_2)\ )$$

where *dist* is the network distance between any two PEs, must be minimized. This is called QD for the *quadratic distance* due to its similarity with the quadratic assignment problem.

For most applications, it is desirable that work be evenly distributed throughout the network. In the absence of prior information about the application, we use the assumption that all problem edges carry an equal amount of traffic and the processing of each message generates an equal amount of work. In this case, an even distribution of the problem graph is achieved by distributing the problem edges such that the variance of the number of problem edges incident on each PE is minimized. Hence, we minimize LV, defined as:

$$\begin{aligned} LV &= var(LD_w), \quad \text{for} \quad w \in V_n, \quad \text{where} \\ LD_w &= \sum_{v \in inverse\text{-}image(w)} deg(v), \quad \text{for} \quad v \in V_p,\ w \in V_n \end{aligned}$$

LV is called the *logical variance* of LD, the *logical degree*, which is the number of logical edges incident on each $w \in V_n$.

As mentioned above, these two metrics are competing and work in opposite directions. The obvious mapping that minimizes QD maps all vertices onto one PE. This produces a summed network distance of zero but is clearly contrary to LV. Conversely, LV can be minimized while QD is not. The problem edges may be evenly distributed while all adjacent vertices are mapped to diametrically opposite sides of the network. To find useful maps, we must control the contribution of each metric by using a weighted composite.

To find a useful weighting, both metrics must be scaled to comparable numerical ranges. To do this, we will identify the minimum, maximum and *mode* values for QD and LV. The minimum QD occurs when all problem graph edges are mapped to one PE. The maximum QD occurs when all communicating problem graph vertices are mapped to diametrically opposite sides of the network. Hence, we have

$$QD_{min} = 0, \quad QD_{max} = |E_p| \cdot diam$$

where *diam* is the diameter of G_n, i.e., the network. The mode of the distribution of the quadratic distance for all possible maps, however, occurs when each problem graph edge

rely on some property of the problem graph or some *a priori* knowledge about the problem itself.

One such property is whether the the problem and network graphs are of equal size such that the vertex map is one-to-one. This can be simpler than when the problem graph is larger than the network graph such that the map is many-to-one. Bokhari [9], for example, represents the problem and network graphs by adjacency matrices and maximizes the number of matching edges, i.e., matching matrix elements, by permuting the mapping of vertices, i.e., the matrix rows and columns. To do circuit board placement, Goto [13] uses generalized force-directed relaxation to iteratively improve the total wire routing length. Fukunaga et al. [11] first use a graph space transform that moves vertices closer together that have heavier edge weights corresponding to higher communication capacity. Then linear assignment is used to assign vertices to the closest PE in a grid.

Another property is whether the problem graph is a *task precedence* or a *task interaction* graph [22]. The first is a directed, possibly cyclic graph that defines the partial order in which operations must occur. Dataflow graphs fall into this category. The second is an undirected graph that defines possible process interaction but contains no information about if or when interaction will occur. If the execution order is known, as in a task precedence graph, problem edges can be partitioned into *phases* that do not overlap in time. Two edges from different phases can then be mapped onto the same network edge since they do not demand bandwidth simultaneously [17]. If the vertices are also weighted proportionally to the work they represent, a *critical path* can be identified that determines which tasks execute in parallel [19].

It is also possible to exploit structural properties of the problem graph. Hong et al. [15] deal with dataflow graphs where every operator (vertex) has either (1) one incoming edge and many outgoing edges, or (2) many incoming edges and one outgoing edge. Tree-shaped partitions of such a graph can be easily defined and mapped onto the faces of an n-cube. Sadayappan and Ercal [22] describe mapping planar finite element graphs onto processor meshes by using 1-D and 2-D strip partitioning. Afterwards, a boundary refinement procedure can be used to improve load balancing. If the problem graph is a mesh with unequal work estimates (vertex weights) in different regions, binary decomposition can be used to recursively partition the mesh into nearly equal work loads [5].

An important property of the mapping technique itself is whether it is *deterministic* or *non-deterministic*. Most of the methods mentioned so far use some form of iterative improvement or local neighborhood search, which are deterministic, to drive their algorithm. The Prep-P system [6], for example, uses local neighborhood search to contract the problem graph to the same size as network graph and then uses the Kernighan and Lin algorithm to assign partitions to PEs. While these algorithms may have reasonable time complexities, the solutions they produce may be trapped in the smallest local minima.

Simulated annealing, however, is a non-deterministic algorithm that can probabilistically escape local minima in search of a global minima [16]. It has been used for a wide variety of NP-Complete combinatorial optimization problems [16, 21, 18] and can produce a many-to-one map without relying on task precedence or any structural properties of the problem graph and can even be executed in parallel [12]. The only requirement is that an energy function must be defined that captures the characteristics to be optimized. This is the topic of the next section.

an n-cube. The problem graph, however, could be very regular, very irregular, or anywhere inbetween, depending on the problem. Of course, when perfect regularity exists, an analytical solution may exist or heuristics may do very well. When perfect regularity doesn't exist, however, how does the relative structure of the two graphs affect the performance of annealing and quenching? We will investigate these questions by mapping four different graph types that represent problem graphs with a range of structure onto a regular network graph.

We begin in Section 2 by introducing the mapping problem and defining an energy metric that satisfies the two competing requirements stated above that are of interest in parallel processing. We also introduce ultrametricity and the triangle statistics used to measure it. In Section 3, we identify four graph types and map these onto a regular network graph using annealing and quenching. The resultant maps are then analyzed using ultrametricity and the correlation between energy and distance. In Section 4, small progressive changes are made in one graph type to demonstrate the effect of changes in graph structure on the observed ultrametricity. Conclusions are given in Section 5.

2 Preliminaries

2.1 The Mapping Problem and Some Techniques

Many models of parallel computation have been developed such as actor-based systems [14, 2], objected-oriented systems [25], or dataflow [10], where the computation is viewed as a graph, where vertices represent independent subtasks and edges represent data dependencies among these subtasks. The ideal architecture to support such a program graph is one where every graph vertex corresponds to a processing element (PE) and every edge corresponds to a communication channel. Of course, ideal architectures are never built. Instead, a finite number of PEs are connected in a very regular topology to form a loosely-coupled, message-passing multiprocessor. Hence, the program or problem graph must be *mapped* onto the network of PEs such that parallelism is maximized yet communication costs are minimized.

Such mappings can either be *static* or *dynamic*. If demands for parallelism or communication change dramatically during a computation in an unforeseeable manner, a system could respond by changing the map on the fly. Of course, doing so incurs a greater runtime cost. In many cases, however, static mapping done at compile-time is sufficient. This is typically the case when the application is a knowledge or database against which large numbers of queries are processed concurrently. The diverse requirement of independent queries tend to spread computation uniformly throughout the entire network. Examples of such systems are object-oriented databases [4], dataflow databases [7], and the message-oriented information management system described in [8].

Finding even a static map is clearly a combinatorial optimization problem that is NP-Complete and thus computationally intractable for graphs of any significant size. Consequently, many approximation techniques have been developed that generate near-optimal solutions in a time much shorter than finding exact solutions. Most of these techniques

1 Introduction

Simulated Annealing is a computational technique of finding a near-optimal "solution" to an optimization problem by making random changes in the solution and probabilistically accepting the change depending on whether it improves or degrades the solution [16, 1]. It is well-known that there is a trade-off between the quality of a solution and the speed with which it is found. A long, slow annealing tends to produce better solutions. A quick annealing, however, called *splat* annealing or *quenching*, can be done is a small fraction of the time and still produce a comparatively good solution. What controls how well annealing and quenching can do on a given optimization problem?

This can be answered by introducing the concept of a *configuration space*. First, an optimization problem can be represented as a set of configurations. The Traveling Salesman Problem, for example, can be represented by the set of all possible routes through the chosen set of cities. Similarly, the Mapping Problem can be represented by the set of all possible ways of mapping one graph onto another. Second, if the set of configurations is taken to be a set of vertices, then edges can be defined between them by the method of making *moves* that change one configuration into another. Third and last, if a *cost* or *energy function* is defined that numerically identifies the quality of any given configuration relative to all others, then the configuration space can be thought of as an *energy landscape*. On an intuitive level, annealing and quenching will do worse in a mountainous configuration space with rough, narrow, steep valleys, i.e., with local minima that are difficult to escape. Better solutions will be found in a rolling configuration space with smooth, wide, shallow valleys.

How can we determine the character of the landscape for a given configuration space? This is very difficult to do; even for relatively small, finite optimization problems, the configuration space can be combinatorially immense. It is just not feasible to compute the size and shape of n-dimensional energy "valleys" where every vertex could have hundreds of incident edges. Alternatively, Solla et al. hypothesize that configuration spaces exhibiting a high degree of *ultrametricity* are more suitable for annealing [23]. Their argument is that a correlation between the energy difference between two configurations and the number of moves or *distance* between them indicates a "topological organization" that is "consistent with that of a uniform tree," a structure that satisfies the ultrametric definition.

Can we use ultrametricity and the correlation between energy and distance to determine if annealing and quenching will perform well on a given class of optimization problems? Concommitantly, how does the "structure" of a class of problems affect the degree of ultrametricity and the performance of annealing relative to quenching? These are the main questions addressed by this paper. They are motivated by our research in parallel processing where a problem graph of communicating tasks must be mapped onto a network graph of processing elements such that (1) the communication distance is minimized, and (2) the problem graph is evenly distributed over the network. Typically, the network graph is very regular, such as a grid or

The Effect of Structure in the Mapping Problem Using Simulated Annealing

Craig Lee
Lubomir Bic
Department of Information and Computer Science
University of California, Irvine

July 23, 1991

Abstract

In parallel computation, a common requirement is the mapping of a problem graph of communicating tasks onto a network graph of processing elements such that (1) the communication distance is minimized, and (2) the problem graph is evenly distributed over the network. Since this is an NP-Complete optimization problem, many mapping techniques have been developed that rely on known properties of the graph or the application problem. Some applications, however, offer few or no such properties that can be exploited. In this case, and when static mapping is sufficient, simulated annealing becomes an attractive alternative. Typically, the network graph is very regular, such as a grid or n-cube. The problem graph, however, could be very regular, very irregular, or anywhere inbetween, depending on the problem. How does the relative structure of the two graphs affect the performance of annealing?

To investigate this question, we used simulated annealing to map four graph types of varying "structure", representing possible problem graphs, onto a graph with a very regular structure, representing a network of processing elements. Many trials were collected for relatively slow annealing and for faster annealing which we call quenching. The overall performance of these trials over all graph types and speeds of annealing is compared by correlating the distance between maps and their energy difference, and by measuring their ultrametric correlation. These results show that higher regularity produces (1) a higher but narrower range of final energy values, (2) a higher distance-energy correlation and (3) a higher degree of ultrametricity. This implies that a regular problem and network graph produces a regular configuration space in which the local minima are ultrametrically distributed. This supports the claim that simulated annealing performs better on configuration spaces exhibiting a higher degree of ultrametricity. Since a higher regularity produced more closely clustered energy values for both annealing and quenching, a shorter annealing schedule can be used with a higher probability that the resulting maps will not be far apart in energy.