Approaches to Physical Optimization

Geoffrey Fox

CRPC-TR91124
April 1991

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

SIAM J. Sci. Stat. Comp.
Vol. 0, No. 0, April 1991

APPROACHES TO PHYSICAL OPTIMIZATION

GEOFFREY C. FOX†

Abstract. Physical Optimization is the use of analogies from nature to solve optimization problems. This approach leads to approximate solutions with time complexities that are much lower than traditional exact methods. The algorithms parallelize straightforwardly and promise the solution of many practical problems on the large scale parallel computers that are expected in the future. We describe and contrast four related physical optimization methods; the well-known simulated annealing and neural network approaches, as well as the important but less well understood deterministic annealing and elastic network methods.

1. Introduction.

Physical Computation encompasses a variety of ideas that can be loosely classified as the use of physical analogies or methods, from the physical sciences to problems outside their normal domain of applicability. One example is the use of simulated annealing (an idea from physics) to chip routing and placement [1] (a problem in optimization). Another is the use of neural networks (an idea from biology) in learning and pattern recognition, which are problems in computer science and robotics.

Note that as used here, "physical" means "pertaining to nature" and is broader than just physics. However, the latter field is particularly relevant as physics has studied large complex systems, albeit those obeying Newton's and other basic laws of physics. For example, in thermodynamics, we find a theory describing very large systems in a way that is insensitive to irrelevant microscopic detail. A key feature of physical computation is a methodology that naturally tackles large problems; we can anticipate a growing role for physical computation as the growing power of computers allows the simulation of larger and larger systems. Traditional methods (for optimization) have time complexities that scale exponentially in problem size, while physical computation is often essentially linear. The factor of a thousand in computer performance improvement, expected by the year 2000, makes little impact on an exponentially complex algorithm; however, it implies a revolution for an algorithm with linear time complexity.

Optimization is an important application of physical computation, and Simic originally introduced the term physical optimization [2]. Indeed, most physics laws can be formulated variationally as an optimization problem, while nature is also involved in optimization. Thus, in the long term, the evolution of the human race is maximizing perhaps some combination of survival and happiness. In the short term, we interpret visual and other sensor information optimally according to our prejudices and experience. These last two analogies lead respectively to genetic [3] and neural net approaches to optimization. Simulated annealing minimizes the (free) energy by Monte Carlo methods; and later we will see elastic net and deterministic annealing approaches to optimization. These

†Syracuse University, Northeast Parallel Architectures Center, 111 College Place, Syracuse, New York 13244-4100.

correspond to non-statistical variational methods from physics applied to optimization. Maximum entropy or information theory leads to similar approaches based on analogies from engineering.

Above, we listed several ideas that we collectively call physical optimization. They can be contrasted with other methods for optimization. Heuristics can be considered as an approach motivated by the problem; combinatorial optimization as one from mathematics, and expert systems as one from computer science.

There is no universally good approach to optimization. Each method has different tradeoffs in robustness, accuracy, speed, suitability for parallelization, and problem size dependence. For instance, neural networks do simple things on large data sets and parallelize easily, while expert systems do complex things on small data sets and parallelize with difficulty. In nature, we see at least four approaches combining to solve the problem of survival. On the long term, a genetic algorithm is used to evolve people to maximize survival. On the short term, we wish to avoid being eaten by a lion. A relatively simple low-level vision network with largely local connections is used to process the initial image. A learning (learned) back propagation like network, may be used to distinguish various animals in the scene. A high level, possibly expert system like reasoning, is used to optimize escape procedures after the lion has been identified.

We also note that physical analogies tend to be fundamentally imprecise; when applied to optimization, they find approximate and hopefully good solutions, but not the best. Combinatorial optimization aims for the exact solution. In practice, approximate solutions to large real world problems are all that is required and, indeed, all that is warranted by imprecise data.

In this paper, we present a unified treatment of four methods of physical optimization; namely, simulated annealing, deterministic annealing, elastic and neural networks. Genetic algorithms are also of great promise and importance, but their formulation is distinct from our four methods and outside the scope of this brief treatise. In the next section, we present a general framework and then describe each method separately. Finally, we review some applications where these ideas might be particularly important. A longer discussion of physical computation will be found in [4] and references cited there.

2. General Formulation of Physical Optimization.

Consider optimization formulated as the minimization of a function E:

$$E = E\,(\text{parameters}\,\underline{y}) \tag{1}$$

where the parameters $\underline{y}$ can be continuous, discrete or a mix. We choose to interpret E as the energy of a system whose states are labeled by the allowed values of $\underline{y}$. Minimizing E is equivalent to finding the ground state of the associated system. We introduce a fake temperature T and set $\beta = 1/T$; in particular cases, we will have a simple physical interpretation of T as, for instance, the scale or granularity at which the problem is formulated. A state of the system is labeled by $\underline{y}$ and to each state we associate a probability

$$\begin{aligned} Pr(\underline{y}, T) &= \frac{e^{-\beta E(\underline{y})}}{Z} \\ \text{where} \quad Z &= \sum_{\underline{y}} Pr(\underline{y}) \end{aligned} \tag{2}$$

As $T \to 0$ or $\beta \to \infty$, the minimum $\underline{y}_{\text{min}}$ of E dominates in Z and the probability of this state tends to unity. The basic idea in annealing is to find the minimum $\underline{y}_{\text{min}}(T)$ of $F(T)$ for $T \sim 0$ by tracking $\underline{y}_{\text{min}}(T)$ from high to low temperatures. Here, $F = E - TS$ is the free energy expressed in terms of original energy E and entropy S. We have some reasons to believe that for many important problems, $\underline{y}_{\text{min}}(T)$ is continuous in T. Then, this continuation technique should avoid local minima, such as the one shown on Figure 1, as it is often easy to find true global minima at high temperatures.

Figure 1: Schematic of the free energy $F(T)$ at a set of temperatures $0 \sim T_1 < T_2 < T_3 < T_4 \sim \infty$.

In the following sections, we describe four distinct methods of finding $y_{\rm min}(T = 0)$ by tracking the system in temperature down from high to low values.

3. Simulated Annealing.

This is the most general and reliable of the four methods discussed in this paper. It was introduced in [1] and has since been applied to all branches of optimization, including the traditional Travelling Salesman Problem. One uses the standard Monte Carlo procedure used extensively to study statistical physics problems, where Equation (1) is a "true" energy function. At each temperature, T, one equilibriates the system and finds rather than $y_{\rm min}(T)$, the sample mean $y_{\rm sample}(T)$ of configurations $\underline{y}$ weighted by $Pr(\underline{y}, T)$. As temperature $T \rightarrow 0$, the minimum and sample mean become identical. Various clever methods, including the Metropolis algorithm and importance sampling, have been developed but this method is computationally costly; very many Monte Carlo steps are needed in the annealing procedure.

We view simulated annealing as the standard of excellence—in terms of minimum finding—by which to judge the other three methods described here. The latter are faster as they avoid costly Monte Carlo steps but less reliable, i.e., they are less successful in avoiding local minima. All the methods have the disadvantage that they cannot, in practice, either guarantee achieving a true global minima or easily estimate the quality, $F(y_{\rm min}$ (found as $T \rightarrow 0)) - F(y_{\rm min}$ (true)), of the solution. Experience has shown that several of these methods give very reliable answers, near to the true minima, for a variety of problems. However, we must remember that all our physical optimization approaches are methods designed to find approximate and not exact minima.

4. Deterministic Annealing.

This describes a class of methods [5] which use heuristic (approximate) minimization methods to find the minimum $\underline{y}_{\rm min}(T)$ of the free energy $F(T)$ by starting with the initial guess, $\underline{y}_{\rm min}(T) = \underline{y}_{\rm min}(T + \delta T)$, as the minimum found at a higher temperature. One could choose an iterative steepest descent method, but any deterministic approach to finding $\underline{y}_{\rm min}(T)$ is possible. One can also mix deterministic and Monte Carlo methods. The latter idea underlies the molecular dynamics approach to Quantum Chromodynamics Lattice Simulations [6], [7].

We will illustrate these ideas by the work of Rose and Gurewitz for clustering [8], [9]. Consider a set of data points, x, which we wish to associate into clusters. This type of problem comes up in many applications, but here we will consider physical clusters in a two-dimensional space such as the 360 points shown in Figure 2 and generated by four clusters [8], [10].

Figure 2. Deterministic Annealing Clustering of four clusters. The lines are the decision boundaries. The final beta is 0.1, and the final energy is 30.05. ○—computed cluster mean. X—center of cluster used to generate points $\underline{x}$ [8].

For each data point x, we assign an energy $E_x(j)$ (cost) for it to belong to the cluster j with mean y_j. We sum over the uninteresting variables that specify the assignment of x to one of the N_c clusters. Then the partition function is

$$Z = \prod_x \sum_{k=1}^{N_c} \exp\left[-\beta E_x(k)\right] \tag{3}$$

and the free energy

$$F = -\frac{1}{\beta} \log Z \tag{4}$$

If the clusters were due to Gaussian fluctuations, then we can take

$$E_x(j) = |x - y_j|^2 \;, \tag{5}$$

and now the cluster centers are determined by the deterministic annealing condition

$$\frac{\partial F}{\partial y_j} = 0 \tag{6}$$

which gives the implicit equation

$$y_j = \frac{\sum_x x \, Pr\,(x \text{ in cluster } j)}{\sum Pr\,(x \text{ in cluster } j)} \;, \tag{7}$$

where y_j also appears on the right hand side from the expression for the probability;

$$Pr\,(x \text{ in cluster } j) = \frac{\exp\,(-\beta|x - y_j|^2)}{\sum_{k=1}^{N_c} \exp\,(-\beta|x - y_k|^2)} \;. \tag{8}$$

We can solve the implicit equation iteratively starting with

$$y_j(T = \infty) = \text{mean of all } x\text{'s} \tag{9}$$

and gradually reduce the temperature T. $y_j(T+\delta T)$ is used as a starting point to find $y_j(T)$. Note that Equation (7) surely has many local minima, but these are avoided by the annealing, as we can rigorously find the global minimum at a high temperature and track it down by lowering the temperature. This particular example has a striking pattern in its temperature dependence. At a high temperature, all the clusters are degenerate with the same y_j. As β is increased, one finds critical temperatures (β) at which the clusters break into two sets—again the members of each set have identical y_j [11].

The basic terms in the exponential are $|x-y_j|^2/T$ and so we see that, for this problem, $T^{\frac{1}{2}}$ can be considered as the length scale at which the system is observed. Thus, for this application, annealing is equivalent to a multiscale approach; we initialize the fine scale optimization with the results of a coarse scale analysis. The utility of this approach is known in many fields, including the well known multigrid method for particle differential equations. It also has been applied to vision by Terzopoulos [12], [13].

The deterministic annealing approach is familiar in Chemistry where particle dynamics and Monte Carlo are both used to find the ground state of a complex molecule. The atoms in the molecule are often found experimentally—say from NMR measurements—and the annealing minimizes a potential containing both physical forces and artificial terms representing agreement of the model with the data. This can be viewed as physical computation addressing the optimization problem "what molecule best fits the experimental data and is consistent with Chemistry" [14].

We expect that deterministic annealing has many other applications and, indeed, we can view the elastic network of Section 6 as a special case of this method. In particular, Rose has shown how to formulate the Travelling Salesman Problem from this point of view [15], [5].

5. Neural Networks.

These were first introduced for the Travelling Salesman Problem (TSP) by Hopfield and Tank [16], [17]. Let us first discuss this case with a set of N cities labeled by the integer $p=1\ldots N$. We wish to visit each city once and once only in a tour that minimizes the total distance traveled. We let $i=1\ldots N$ label successive steps of the tour with $p=P(i)$ labeling the city visited at the i'th step. Then we need to minimize

$$\sum_{i=1}^{N-1} d_{P(i)P(i+1)} \tag{10}$$

This assignment ($i \rightarrow P(i)$) is the classic NP complete discrete optimization problem [19] which is often used as the standard benchmark for discrete optimization methods. The following examples make it clear that the TSP is not necessarily typical of all such problems, and methods that perform poorly on the TSP work well on other (NP complete) optimization problems.

We introduce the neural variables

$$\begin{aligned} \eta_p^i &= 1 \ \text{ if } \ p=P(i) \\ &= 0 \ \text{ if } \ p \neq P(i) \end{aligned} \tag{11}$$

and we rewrite Equation (10) as

$$E_1 = \sum_i \sum_{p,q} d_{pq}\, \eta_p^i\, \eta_q^{i+1} \tag{12}$$

where in Equation (10) and Equation (12), d_{pq} is the distance between cities p and q. We now have a nice quadratic form to minimize as a function of the N^2 neural variables η_p^i. Unfortunately, not all choices of η_p^i are allowed; for this to correspond to a true assignment, one needs to satisfy constraints that each i corresponds to one p and vice versa. These can be written as

$$\begin{aligned} &\sum_i \eta_p^i\, \eta_q^i = \delta_{pq} \\ &\sum_p \eta_p^i\, \eta_p^j = 0 \quad i \neq j \end{aligned} \tag{13}$$

This is implemented by minimizing

$$E = E_1(\text{equation 12}) + \sum [\text{constraints—equation 13}] \tag{14}$$

with penalty terms, simple linear or quadratic functions of the forms (Equation (13)), which are positive when constraints are violated.

Combining Equation (14) with the physical optimization framework of Section 2 leads to a traditional statistical physics problem with N^2 "spins" η_p^i governed by an energy function E. The resultant simulated annealing or Monte Carlo approach to this statistical physics formulation does provide an effective approach to the TSP [20]. Here we will study a faster, but less reliable, deterministic method. A well-known approximate method for studying such physics systems is the mean field approximation. Consider an equation such as

$$\langle \eta_r^k \rangle = \sum_{\text{states}} \eta_r^k \exp\left(-\beta E(\eta_1^1 \dots \eta_r^k \dots \eta_N^N)\right)/Z \tag{15}$$

Then one can calculate this if one linearizes the exponential by approximating any term in E that is quadratic in η by a linear dynamic term multiplied by the "mean field"—the other η's replaced by their mean value. Roughly, one substitutes

$$\underline{\eta} M \underline{\eta} \rightarrow \langle \underline{\eta} \rangle M \underline{\eta}. \tag{16}$$

With the approximation of Equation (16), one can sum over the dynamical variables labeling the states, and Equation (15) can be converted into a deterministic equation for $\langle \eta_r^k \rangle$. At the desired ground state, $\eta_r^k = \langle \eta_r^k \rangle$ and one finds a deterministic method for finding the minimum of E.

As shown in [21], this method is not very effective for the TSP, as we are only guaranteed approximate solutions. These are both non-optimal in E_1, Equation (12), but also have nonzero values for the constraint penalty terms in Equation (14). Thus, the derived values for η_p^i do not correspond to a real "tour". So far, no satisfactory way has been found for converting these illegal solutions into viable ones, and so the neural network approach is currently unsatisfactory for even modest $N \geq 50$ cases where other methods, including both combinatorial optimization and even simulated annealing approaches, are excellent.

However, we believe that this negative assessment is not generally true, and neural networks are effective for some NP complete optimization. In particular, we successfully used neural networks to solve the optimization problems associated with the static and dynamic load balancing for large scale parallel computers [22], [23]. This can be formulated similarly to Equation (11) to Equation (16) with a different form for E, which now represents total parallel computer execution time, including the effects of communication and load imbalance [4]. We must solve an association problem, $m \rightarrow P(m)$ where P is the label of the processor to which process (data point, unit of work) m is assigned. If we used neural variables, $\eta(m,p)$, analogous to Equation (11) such that $\eta(m,p) = 1$ if $p = P(m)$ and zero otherwise, then we would find the same unsatisfactory results as for the TSP. Rather, we write for M work units indexed by m, and N processors indexed by p.

$$P(m) = \sum_{k=0}^{d-1} 2^k \eta_k(m) \tag{17}$$

and the $Md = M \log_2 N$ neural variables $\eta_k(m)$ provide a non-redundant specification of the decomposition. This is to be compared to the MN variables $\eta(m,p)$ in the redundant formulation. Defining $w(m)$ as work associated with m and $C(m,m')$ as communication needed between m and m', then we can now specify the energy of the associated physical system as

$$\begin{aligned} E &= E_{\text{calc}} + E_{\text{comm}} \qquad \text{where} \\ E_{\text{calc}} &= \frac{1}{N} \sum_{m,m'} w(m)\, w(m') \prod_{k=0}^{d-1} [1 + s_k(m) s_k(m')] \\ E_{\text{comm}} &= \frac{1}{4} \sum_{m,m'} C(m,m') \sum_{k=0}^{d-1} [1 - s_k(m) s_k(m')] \end{aligned} \tag{18}$$

with spins $s_k(m) = 2\eta_k(m) - 1$ taking values of ± 1 (for $\eta_k(m) = 1$ or 0). The physical analogy is particularly good here with E_{calc} (balancing computation on each node) as a short range repulsion and E_{comm} (minimizing communication) as a long range attractive force.

Equation (18) can now be used in the physical optimization approach; both the simulated annealing and neural network methods can be applied [24]. Indeed, even in simulated annealing, the elegant neural network choice of variables is preferred to a direct expression of energy E in terms of $P(m)$. As shown in [22], the mean field method developed by Hopfield and Tank for the TSP is directly applicable and gives excellent results. These are comparable in quality to simulated annealing, but much faster as one is just solving deterministic equations. As already mentioned, neural networks work well here, as the objective function E in Equation (18) has no penalty terms.

The neural representation of Equation (17) was originally motivated by the hypercube topology. However, it is generally useful and there is an interesting analogy with the clustering method in Section 4. Thus, the neurons provide a multiscale representation with $\eta_{d-1}(m)$ the coarsest and $\eta_0(m)$ the finest detail. From this point of view, it is clear how to generalize this approach to non-hypercube topologies by using the appropriate hierarchical neural multiscale representation in "processor space".

6. Elastic Network.

This approach was introduced for the TSP by Durbin and Wilshaw [18], [25] as a physically based model that outperformed the neural network in this case. However, we will first describe it for the multi-object navigation problem. This occurs in routing messages in a switch or communication network; dynamic load balancing; robot arm manipulation and in the navigation of one or more vehicles in a physical terrain. In each case, one needs to find the paths of several objects subject to various constraints; these reflect lack of collision between objects, terrain features and object movement characteristic. An idealized situation is chosen in Figure 3 for two vehicles in a two-dimensional terrain with sundry impenetrable obstacles. The vehicles need to travel from a given source to a destination in minimum time [26], [27]. The methods discussed generalize straightforwardly to many vehicles, arbitrary sources and destinations, and general movement and terrain characteristics.

Figure 3. The Physical Computation picture for navigation in the generalized elastic net approach [28].

A traditional approach to this problem discretizes each of the l degrees of freedom into N cells, and has a time complexity of $O(N^l)$ [29]. This is practical and probably the best approach for small values of l, e.g., single vehicle navigation [30], [31], but is impractical for large l. We originally used a neural network approach to this problem and although interesting results were obtained, we found difficulties similar to those discussed in the last section for the TSP [32]. We obtain the elastic network by choosing a better set of variables which have fewer constraints. Such a set is formed by the positions $\underline{x}_i(t_j)$ of vehicle i at a discrete set of time values t_j. We represent these as the positions of the beads $B_{i,j}$ shown in Figure 3. We can represent the goal and constraints as an energy function E for these beads.

The goal of minimum travel time translates into elastic forces between neighboring beads on each path. Other constraints correspond to avoidance of obstacles—this is represented as repulsive forces between obstacles and beads; the avoidance of collisions corresponding to repulsive forces between the beads specifying different vehicles at each time step. The formalism allows the inclusion of many different issues such as variable terrain and vehicles of finite size, and these are discussed in [28].

One cannot easily minimize the full energy function, but we use the annealing idea by starting off at high temperature, T, with the difficult constraints (repulsive forces in Figure 3) switched off.

$$E = E_1\,(\text{elastic}) + \frac{1}{T} E_2\,(\text{constraints}) \tag{19}$$

One can minimize E_1 exactly and one would find, in a simple case, the solution as the straight line between source and destination. We now apply the deterministic annealing idea already described in Section 4. $\underline{x}_{i,\,\text{minimum}}(t_j, T)$ are found by minimizing Equation (19) starting with the estimate

$$\underline{x}_{i,\,\text{minimum}}(t_j, T) = \underline{x}_{i,\,\text{minimum}}(t_j, T + \delta T) \tag{20}$$

As illustrated in Figure 1, one expects to avoid local minima by tracking down from the high temperature where you can guarantee to find the global minimum.

We have made a preliminary investigation of these ideas, and the results are very promising and have significant advantages over other techniques. Deterministic annealing is fast and avoids local minima.

We now return to the TSP which can be formulated in a similar way. Consider the TSP for a set of cities in a two-dimensional space. Suppose the salesman is at position $\underline{x}_i$ at time step $i = 1 \ldots N$ where one has N cities. We wish to find a tour of minimum length where each $\underline{x}_i$ is located at the position of a distinct city.

Again, one places a bead at each position $\underline{x}_i$ and forms an energy function for the N beads with two terms. The elastic force between beads gives a potential corresponding to goal of minimizing tour length. The second term is a set of forces between cities and beads which is minimized when each bead is coincident with a distinct city.

$$E = E_1\,(\text{elastic} = \text{tour length}) + \frac{1}{T} E_2\,(\text{minimized when cities and beads coincide}) \tag{21}$$

At $T = \infty$, the beads collapse to a zero length tour at the centroid of the cities. As T is decreased, we increase the importance of constraint E_2 and as $T \to 0$, one recovers the proper TSP. In practice, this approach is successful and competitive with other approaches.

A beautiful paper by Simic [2] clarified the relation between the neural and elastic networks for the TSP. He was able to derive Equation (21) from the neural network, Equation (14), by using the mean field approximation to a set of degrees of freedom which satisfy some of the constraints in Equation (13). Thus, we find both neural and elastic networks as similar methods, but with different choice of degrees of freedom. The better performance of the elastic network corresponds to a better and less redundant choice of variables. We now can understand the good performance of neural networks in load balancing where they were the natural choice of variables as they needed *no* constraint terms in the energy function.

7. Conclusions.

Physical optimization is a general approach which naturally parallelizes and can be effectively applied to very large problems. It is applicable to continuous and discrete optimization; the discussion of the Travelling Salesman Problem shows how it can be used successfully in NP complete discrete optimization. However, the physical analogies are particularly powerful when one can choose degrees of freedom which are natural for both the original problem and the physical analogy. Examples are the neural network approach to load balancing parallel computations, and the elastic "string of beads" for navigation problems. We expect the latter to be very powerful to navigation and robot manipulation in cases where large numbers of degrees of freedom make alternative approaches less effective. We have shown how the same ideas can be used in track finding and other image analysis applications [4], [33]. They can also be immediately applied to scheduling problems.

Many of these ideas are still poorly developed, but they will grow in importance as problems get larger and computers get more powerful.

8. Acknowledgements.

This work was supported by the Department of Energy: Applied Mathematical Sciences—Grant: DE-FG03-85ER25009; ASAS Program Office Techbase Program, and the National Science Foundation under Cooperative Agreement No. CCR-8809165—the Government has certain rights in this material.

The collaborations with Wojtek Furmanski, Amar Gandhi, Eitan Gurewitz, Ken Rose, and Petar Simic were essential in crystallizing my ideas in this area. I thank TRW for very helpful discussions about robot manipulation.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, *Optimization by Simulated Annealing*, Science, 220 (1983), pp. 671-680.

[2] P. Simic, *Statistical Mechanics as the Underlying Theory of "Elastic" and "Neural" Optimizations*, Network, 1 (1990), pp. 89-103.

[3] M. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.

[4] G. C. Fox, *Physical Computation*, submitted to the International Conference on Parallel Computing: Achievements, Problems and Prospects (1990).

[5] K. Rose, *Deterministic Annealing, Clustering, and Optimization*, PhD Thesis, California Institute of Technology (1990).

[6] S. A. Gottlieb, W. Liu, D. Toussaint, R. L. Renken, and R. L. Sugar, *Hybrid-Molecular-Dynamics Algorithms for the Numerical Simulation of Quantum Chromodynamics*, Phys. Rev. D, 35 (1987), pp. 2531.

[7] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, *Hybrid Monte Carlo*, Physics Letters B, 195 (1987), pp. 216-220.

[8] K. Rose, E. Gurewitz, and G. C. Fox, *A Deterministic Annealing Approach to Clustering*, Pattern Recognition Letters 11(9) (1990), pp. 589-594.

[9] K. Rose, E. Gurewitz, and G. C. Fox, *A Nonconvex Cost Optimization Approach to Tracking Multiple Targets by a Parallel Computational Network*, in D. W. Walker and Q. F. Stout, editors, *The Fifth Distributed Memory Computing Conference, Volume I*, pages 78–84, 1990. IEEE Computer Society Press. Held April 9–12, Charleston, South Carolina. Caltech Report C3P-875.

[10] K. Rose, E. Gurewitz, and G. C. Fox, *Vector Quantization by Deterministic Annealing*, submitted to the IEEE Transaction on Information Theory (1990).

[11] K. Rose, E. Gurewitz, and G. C. Fox, *Statistical Mechanics and Phase Transitions in Clustering*, Physical Review Letters 65(8) (1990), pp. 945-948.

[12] D. Terzopoulos, *Image Analysis Using Multigrid Relaxation Methods*, IEEE Trans. Pattern Analysis Machine Intelligence 8 (1986), pp. 129.

[13] R. Battiti, *Real-Time Multiscale Vision on a Two-Dimensional Mesh of Processors*, accepted for publication in Concurrency: Practice and Experience (1991).

[14] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations*, J. Comput. Chem. 4(2) (1983), pp. 187-217.

[15] K. Rose, E. Gurewitz, and G. C. Fox, *Constrained Clustering as an Optimization Method*, submitted to the IEEE Transactions on Pattern Analysis and Machine Intelligence. Caltech Report C^3P-919 (1990).

[16] J. J. Hopfield, and D. W. Tank, Biol. Cybern. 52 (1985), pp. 141-152.

[17] J. J. Hopfield, and D. W. Tank, *Computing with Neural Circuits: A Model*, Science 233 (1986), pp. 625.

[18] R. Durbin, and D. Wilshaw, *An Analogue Approach to the Travelling Salesman Problem using an Elastic Net Method*, Nature 326 (1987), pp. 689-691.

[19] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.

[20] O. Martin, S. W. Otto, and E. Felten, *Large-Step Markov Chains for the Traveling Salesman Problem*, submitted to J. Complex Systems (1991); Caltech Report C3P-836b.

[21] G. V. Wilson, and G. C. Pawley, *On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank*, Biol. Cybern. 58 (1988), pp. 63-70.

[22] G. C. Fox and W. Furmanski, *Load Balancing Loosely Synchronous Problems with a Neural Network*, in the Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications, Volume 1, G. C. Fox, ed., ACM Press, 11 West 42nd Street, New York, New York 10036, 1988, pp. 241-278.

[23] G. C. Fox, *A Review of Automatic Load Balancing and Decomposition Methods for the Hypercube*, in Numerical Algorithms for Modern Parallel Computer Architectures, M. Schultz, ed., Springer-Verlag, 1988, pp. 63-76.

[24] R. D. Williams, *Performance of Load Balancing Algorithms for Unstructured Mesh Calculations*, to be published in Concurrency: Practice and Experience (1991).

[25] A. Durbin, R. Szeliski, and A. Yuille, *An Analysis of the Elastic Net Approach to the Travelling Salesman problem*, Neural Computation 1 (1989), pp. 348.

[26] A. Gandhi, and G. C. Fox, *Solving Problems in Navigation*, Syracuse University Technical Report, SCCS-9 (1990).

[27] G. C. Fox, *Applications of the Generalized Elastic Net to Navigation*, California Institute of Technology Technical Report, C^3P-930 (1990).

[28] A. Gandhi, and G. C. Fox, *Physical Optimization for Navigation and Robot Manipulation*, Syracuse University Technical Report, SCCS-43 (1990).

[29] G. Heinzinger, P. Jacobs, J. Canny, and B. Paden, *Time-Optimal Trajectories for a Robot Manipulator: A Provably Good Approximation Algorithm*, in Proceedings of 1990 IEEE Robotics and Automation Conference, 1990.

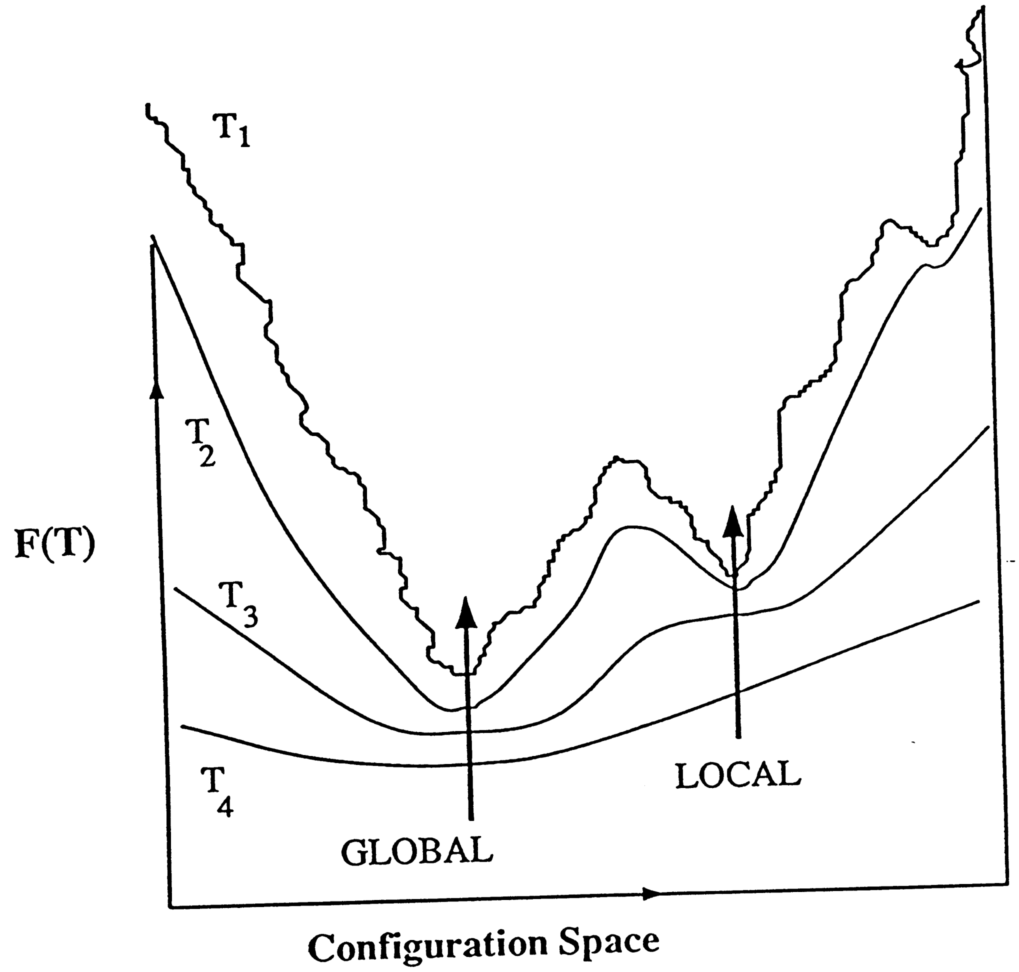
[30] E. Gurewitz, G. C. Fox, and Y.-F. Wong, *Parallel Algorithm for One and Two-Vehicle Navigation*, in D. W. Walker and Q. F. Stout, editors, *The Fifth Distributed Memory Computing Conference, Volume I*, pages 140–147, 1990. IEEE Computer Society Press. Held April 9–12, Charleston, South Carolina. Caltech Report C3P-876.

[31] S. T. Jones, *Solving Problems Involving Variable Terrain, Part I: A General Algorithm*, BYTE (1980).

[32] G. C. Fox, E. Gurewitz, and Y.-F. Wong, *A Neural Network Approach to Multi-vehicle Navigation*, in D. W. Walker and Q. F. Stout, editors, *The Fifth Distributed Memory Computing Conference, Volume I*, pages 148–152, 1990. IEEE Computer Society Press. Held April 9–12, Charleston, South Carolina. Caltech Report C3P-910.

[33] G. C. Fox, *A Note on Neural Networking for Trackfinding*, California Institute of Technology Technical Report, C^3P-748 (1989).

Figure 1



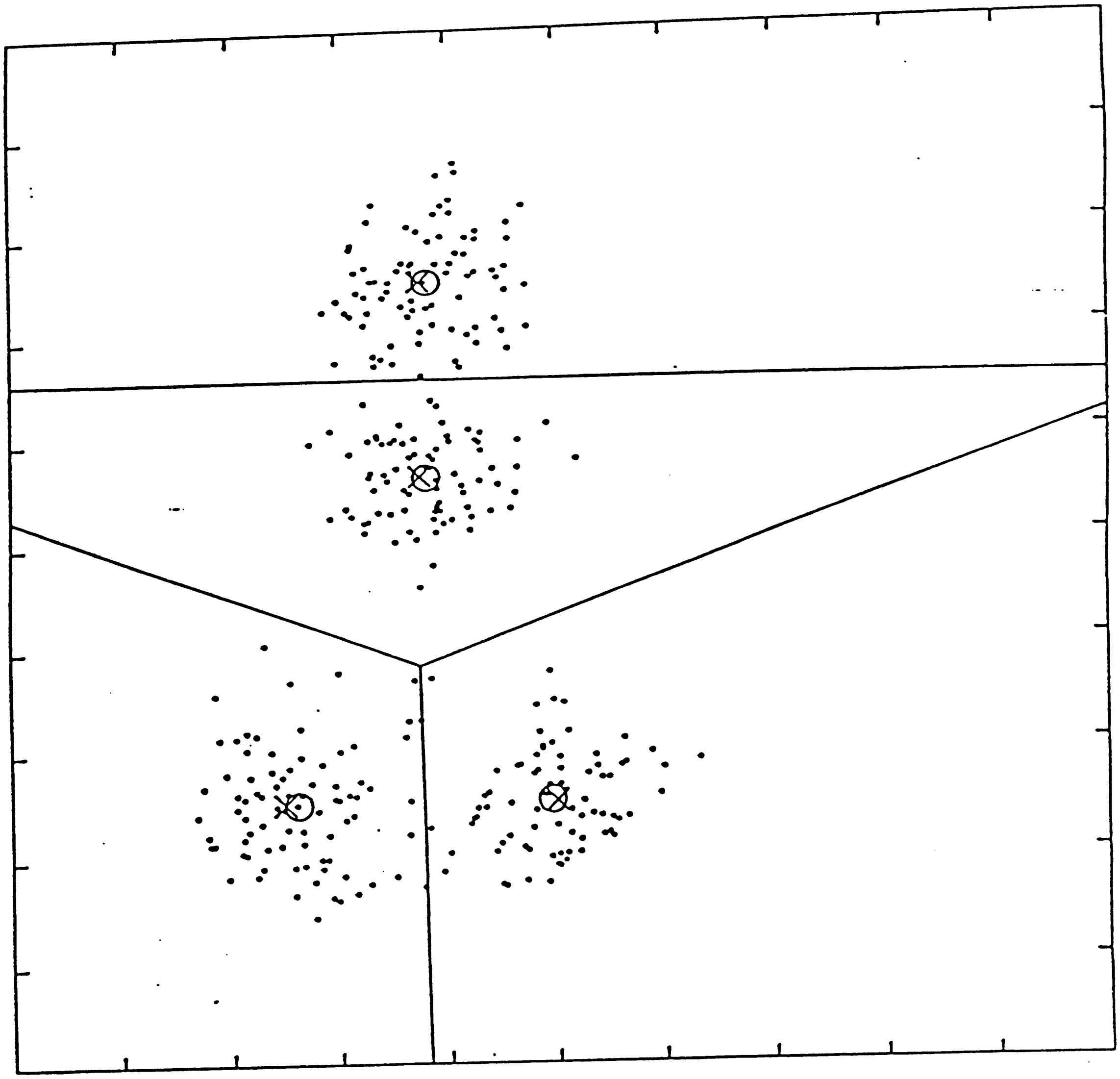
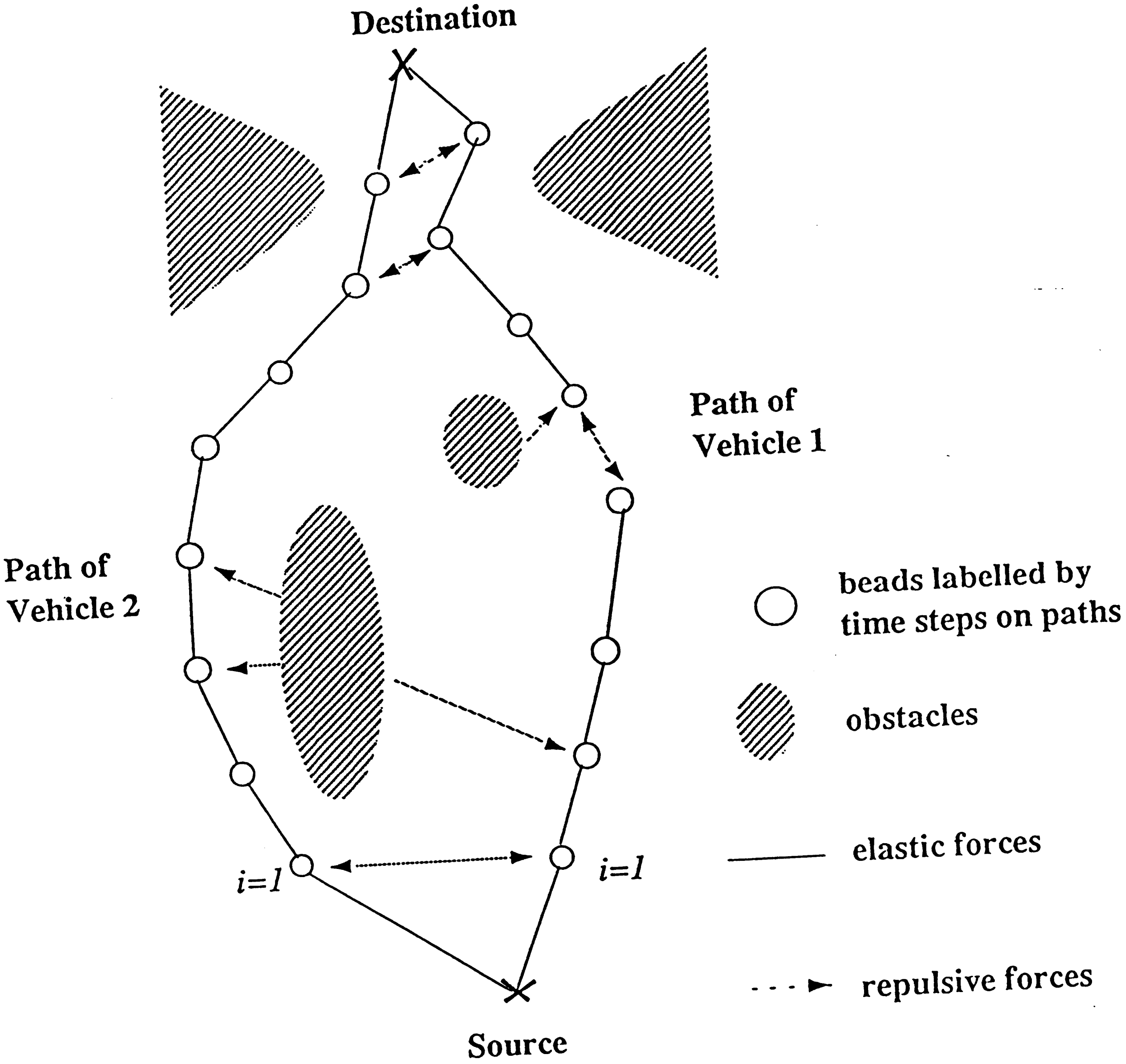


figure 3


Destination
Path of Vehicle 1
Path of Vehicle 2
i=1
i=1
Source
beads labelled by time steps on paths
obstacles
elastic forces
repulsive forces