HEURISTIC *versus* STATISTICAL PHYSICS APPROACH TO OPTIMIZATION PROBLEMS*

C. JĘDRZEJEK† AND L. CIEPLIŃSKI

EFP — The Franco-Polish School
of New Information and Communication Technologies
Mansfelda 4, 60-854 Poznań, Poland

(Received April 6, 1995)

Optimization is a crucial ingredient of many calculation schemes in science and engineering. In this paper we assess several classes of methods: heuristic algorithms, methods directly relying on statistical physics such as the mean-field method and simulated annealing; and Hopfield-type neural networks and genetic algorithms partly related to statistical physics. We perform the analysis for three types of problems: (1) the Travelling Salesman Problem, (2) vector quantization, and (3) traffic control problem in multistage interconnection network. In general, heuristic algorithms perform better (except for genetic algorithms) and much faster but have to be specific for every problem. The key to improving the performance could be to include heuristic features into general purpose statistical physics methods.

PACS numbers: 02.60.Pn, 05.90.+m, 42.79.Ta

1. Introduction

Optimization is a basic numerical procedure. In a famous book *Numerical Recipes* [1] it occupies one chapter. Optimization methods are applied to mostly two classes of problems: (1) minimization of general functions of n variables (for testing purposes functions possessing nasty minima landscape are used) (2) combinatorial optimization: finding the extremum of a function given set of explicit constraints. There exist standard methods

* Presented at the VII Symposium on Statistical Physics, Zakopane, Poland, September 22–28, 1994.

† also Institute of Physics, Jagellonian University, ul. Reymonta 4, 30-059 Kraków, Poland.

such as quasi-Newton, downhill simplex and conjugate gradient. Gradient methods suffer from a common problem; they stuck in local minima.

Physicist came up with several methods having statistical physics origin that claimed to having found a remedy to this problem. One such method, simulated annealing is very popular among physicists [2–4]. For example it has been successfully used to design complex integrated circuits. The arrangement of several hundred thousand circuit elements constituting an integrated circuit has been optimized (this is the number mentioned in Ref. [1], however, in the original work [3] a circuit with only 98 elements was analyzed) more efficiently compared to standard (presumably, heuristic) methods. Unfortunately, this result does not seem to be general. There exists a huge consensus gap between physicists, and computer science and operation research communities, who consider simulated annealing as painstakingly slow and moreover giving results of lower quality. The authors have the statistical physics background, but recently undertook information science problems. Our goal is to give an assessment of statistical physics methods to optimization problems and operation research, in particular to three types of problems: (1) the Travelling Salesman Problem [5], (2) vector quantization [6], and (3) traffic control problem in multistage interconnection network [7].

Before in the literature there were two excellent works by Rujan [8] and Lister [9] with extensive comparison of heuristic and statistical physics optimization results. This review utilizes the results of these works but for larger class of systems and larger systems.

There exists a huge literature on yet another branch of optimization (largely ignored by physicists), stochastic approximation (see, the excellent work by Styblinski and Tang [10]), which will not be discussed in this review.

2. Travelling salesman problem

The Travelling Salesman Problem (TSP) [5] is the Ising model of combinatorial optimization. For combinatorial optimization problems both the configuration states and the constraints are integers. The most important problems in this field belong to the class of NP-complete, *i.e.* time of their solution grows exponentially with the number of elements. Let $K_N = (V_N, E_N)$ be a complete (having every pair of nodes ij connected by an edge e_{ij}) undirected graph with N nodes and $N(N-1)$ edges. The TSP can be stated as follows: for a Euclidean graph find a Hamiltonian cycle (tour that visits each city exactly once) such that it is the shortest for a given distribution of distances between N cities (nodes). Although the exact solutions are known for certain class of problems up to 2392 cities (this largest system was solved by Padberg and Rinaldi [11]), for large N

the computational cost becomes prohibitive. In the need for acceptable although not exact solutions statistical physics methods play a significant role. Various methods are in use including the neural nets of the Hopfield [12] or Kohonen [13] type [14], elastic net [15], simulated annealing [2, 3, 16] or the mean-field. In this work we concentrate on comparison of several approaches that claimed relative success in optimizing the TSP. We mostly benchmark the methods with the use of the popular 318-city TSP, lin318, for which the best known cost is 41345 [17–19].

2.1. Heuristics

The simplest heuristic methods are 2-opt and 3-opt. A 2-opt move consists in cutting 2 links (ij and kl) and replacing them with links ik and jl. This takes $O(N^2)$ time. The 3-opt move consists of breaking 3 links and joining the nodes (they are 8 ways to do this). In general k-opt involve cutting k links and reconnecting them which takes $O(N^k)$ time. In the most recent excellent review by Jünger *et al.* [20], on their sample of test problems, the 2-opt method gives average distance from the exact solution of 8.3% (9.5% on the pr2392 problem [11] in 0.4 s on SUN SPARCstation 10/20). With a restricted version of the 3-opt Jünger *et al.* [20] achieved an average quality of 3.8% (18 s on pcb442). For the TSP problem the best heuristic procedure is the Lin-Kernighan rearrangement (LK) [17]. It builds complicated modifications that are composed of simpler moves, but allowing that some component moves do not necessarily go only down-hill. There are very few implementation of the original Lin-Kernighan procedure [17]. Restricted version of the Lin-Kernighan rearrangement (Jünger *et al.* [20]) give an average distance from the exact solution of 1.5 – 1.8% on the set of problems. For the pr2392 problem the result is 2.5 – 2.95% in 61.7 – 122.3 s on SUN SPARCstation 10/20. Extending the LK procedure with 4-opt and reapplying the heuristics brings the result to 1.75% for the pr2392 problem and increases the time to over 8500 s. The latest procedure is capable of achieving the global minimum for several standard cases, but none above 227 cities. For the lin318 case the error is 0.53%.

2.2. Neural networks: analysis of the Hopfield and Tank type Hamiltonians

Application of neural networks to optimization problems started with the Hopfield and Tank [12] who proposed the following energy function

$$E_{\mathrm{HT}} = \frac{A}{2} \sum_{x,i \neq j} n_{xi} n_{xj} + \frac{B}{2} \sum_{i,x \neq y} n_{xi} n_{yi} + \frac{C}{2} \left(\sum_{x,i} n_{xi} - N \right)^2 + \frac{D}{2} \sum_{x,y,i} d_{xy} n_{xi} (n_{y,i+1} + n_{y,i-1}), \tag{1}$$

where n_{xi} is a peusdospin (occupation) variable for visiting a city x in the n-th step, d_{xy} is the distance between city x and city y, in the last term the i variable is modulo N. The first (second) term is zero if there is no more than one 1 in each row (column). The third term is zero if there are exactly N 1's, and the last term is equal to the cost of the tour. A, B, C, D are positive, but otherwise not precisely defined. Hopfield and Tank used $A = B = 500$, $C = 200$ and $D = 500$ in their work. In spite of initial claims the method generally failed for $N > 10$. Various remedies were proposed.

Recently Mehta and Fulop [21], proposed the following Lapunov function that removed some problems inherent in the original Hamiltonian of Hopfield and Tank [12]

$$E_{\mathrm{MF}} = \frac{H}{2} \sum_{x,i \neq j} n_{xi} n_{xj} + \frac{V}{2} \sum_{i,x \neq y} n_{xi} n_{yi} - G \sum_{x,i} n_{xi} + \frac{D}{2} \sum_{x,y,i} d_{xy} n_{xi} (n_{y,i+1} + n_{y,i-1}) + \frac{S}{2} \sum_{x,i} n_{xi} n_{xi}. \tag{2}$$

More precisely, this energy was used to the Hamiltonian cycle problem which is the special case for TSP. It can be transformed into TSP by completing the connection graph to fully connected graph and ascribing to the original edges weights 0 and to new edges weights 1. Problem is defined by specifying the cost matrix d_{xy} and average connectivity between cities c.

In contrast to the Hopfield and Tank parameterization, Mehta and Fulop also gave relation for coefficients that produce stable solutions. For $G = 1.0$, $V = H = 0.7$, $S = 0.6$ and $D = 0.05 - 0.5$ they achieved 100% of legal solutions.

Among the Hopfield-type Hamiltonians, probably the Potts representation proposed by Peterson and Södeberg [22, 23] enjoys the best reputation

$$E_{\mathrm{PS}} = -\frac{\beta}{2} \sum_{x} \sum_{i} n_{xi}^2 + \frac{\alpha}{2} \sum_{i} \left(\sum_{x} n_{xi} \right)^2 + \sum_{x,y} \frac{d_{xy}}{2} \sum_{i} n_{xi} (n_{y,i+1} + n_{y,i-1}). \tag{3}$$

The advantage of this approach is that the Potts variable exactly fulfills the legality condition for the i-th variable

$$\sum_i n_{xi} = 1 . \tag{4}$$

The corresponding mean-field equation reads

$$\bar{n}_l = \left[1 + \exp\left(-\frac{\partial E[\bar{n}]}{k_B T \partial \bar{n}_l}\right)\right]^{-1}, \tag{5}$$

where l denotes two indices x, i, k_B is the Boltzmann constant, T temperature, and $\bar{n}_l$ is an average population at site l.

For d_{xy} normalized to unit square, and parameter values $\alpha = 1.0$ and $\beta = 0.5$ one rarely obtains 100% of legal solutions (for large problem size) but these solutions are not too far from optimal so hybrid approach with the use for example of *greedy heuristics* or the 2-opt does not make the cost function considerably worse.

Recently, Ciepliński and Jedrzejek [24] have derived that the Ising-type Mehta and Fulop Hamiltonian [21] and the Peterson–Södeberg Hamiltonian [22] have effectively the same form if the following equivalences are used

$$D \longrightarrow 1.0, \quad H \longrightarrow \beta, \quad V \longrightarrow \alpha, \quad G \longrightarrow \alpha, \quad S \longrightarrow \alpha .$$

As a consequence one can compare the methods of solutions of the same effective Hamiltonian.

As for values of coefficients scaling the Mehta and Fulop Hamiltonian by taking $D = 1.0$ and multiplying the rest of coefficients by $10/7$ we obtain

$$H = 1.0 = 2\beta, \quad V = 1.0 = \alpha, \quad G = 1.3 \approx \alpha, \quad S = 0.86 \approx \alpha .$$

The largest difference (by a factor of 2) appears in the first term. The difference in the third term is less important because it is dropped from the Potts Hamiltonian, anyway. We have verified that using $\beta = 1$ for the Potts Hamiltonian makes the results worse.

We start describing results obtained using various methods by comparing different implementations of the same method. We first compare our results for the mean field Potts model with the Södeberg and Peterson results for 50 and 100 cities with $\alpha = 1, \beta = 0.5$, Fig. 1. They are approximately of the same quality (strictly, they are better than in Södeberg, Peterson [22] because of breaking of the symmetry of the direction of a tour and taking $n_{x1} = 1$ for one selected city x, but little worse than in the Peterson work [23], where only one distribution of cities was used). Our calculations using the mean-field method for the Potts representation for the 318 city

problem are more effective (average cost 58000, the best 55000) than the Ising model neural network implementation by Mehta and Fulop [21] (the cost from 64552 to 61337). However, when we recalculated the Ising scheme we found that decreasing the rate of annealing, *i.e.* making the λ parameter smaller than 0.01 (as used by Mehta and Fulop, see his section 8.1.5) brought the average cost to roughly 58000 without significant increase of the computation time. It would be interesting to investigate why the Potts scheme still ends with illegal matrices using the mean-field approach while Mehta and Fulop reported 100% of legal matrices verified also by us.

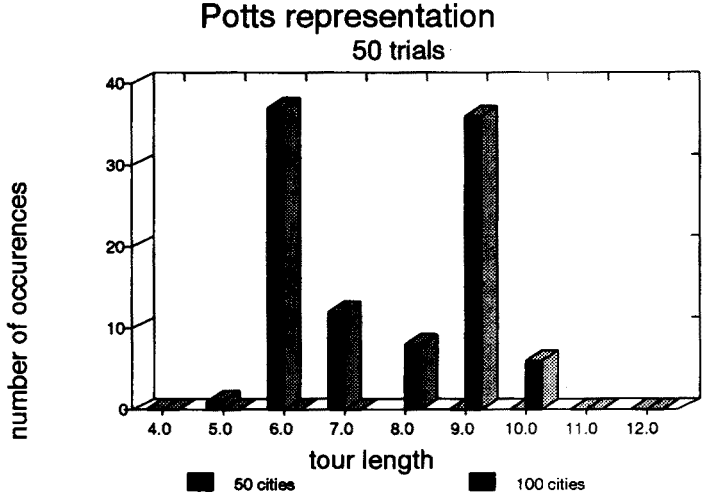


Fig. 1. Histograms based on 50 experiments for 50 and 100 cities TSP.

The choice of an activation function has still not been systematically investigated. We have verified (this observation coincides with that of Mehta and Fulop) that the linear activation function performs much better that sigmoid. Surprisingly, as we verified, radial basis functions gave worse results than linear activation function.

Among neural networks approaches, perhaps the best method is derived from the Kohonen self-organizing feature maps [13] in the Angeniol *et al.* [14] implementation. A close net consisting of M nodes is evolved in such a manner that for a given city i the node closest to the city moves toward it dragging its neighbors albeit with decreasing intensity. A mechanism is provided for starting from a single node in a net up to $M = N$, the number of cities, by duplicating nodes if they are the closest for more than one city in a survey and dropping a node if it is closest to no city in three consecutive iterations. The method is attractive because it is continuous but not in occupation variables but in positions of nodes. It is fast (requires $O(N^2)$

operations), moreover its performance together with the performance of the elastic net [15] rates best among the non-heuristic methods. This can be indirectly inferred from comparison of the performance of the elastic net [15] with the Angeniol *et al.* [14] implementation of self-organizing feature map, and performance elastic net against several methods including the mean field and Hopfield neural networks [22].

2.3. Genetic algorithms

Genetic algorithms involve operations on strings of bits with copying and swapping partial strings as basic operations. A simple genetic algorithm is composed of three operations: reproduction according to values of objective function, crossover and mutation. Genetic algorithms are increasingly popular for solving classical optimization problems [25–28], and different solutions have been proposed concerning the representation of each individual, the size of the population, the crossover and mutation operators, and the initialization strategy. Also hybrid approaches are used with the goal of including heuristic techniques into the pure genetic algorithm schemes. The exhaustive investigation of various modifications of a genetic algorithm in application to the TSP was performed by Prinetto *et al.* [29]. These researchers compared 4 types of crossovers and found the heuristic one is most the efficient. (The heuristic crossover (Grefenstette *et al.*, [25]) is as follows. Starting city for the tour is randomly chosen; then the next visited city is nearest one along one of the parent tours that has not yet been visited in the offspring tour.) Prinetto *et al.* found small effectiveness of mutation operators (purely random changes) and therefore neglected them in presenting results. The value of the Prinetto *et al.* work is somewhat decreased by a large number of misprints in their paper. We therefore, performed our own simulations.

In the coding strategy we used 2 representations: (1) path representation for initialization (Grefenstette *et al.* [25] defined as: the tour is described by a vector of N integers, whose i-th element holds the value j if the city j is reached at the i-th step; (2) adjacency representation after initialization (Grefenstette *et al.*, [25]) defined as: the tour is described by a vector of N integers, whose i-th element holds the value j if i precedes j in the tour. In most simulations we used modified version of the heuristic operator, in which the choice between the two possibilities is made at random with weights inversely proportional to the corresponding distances (Pal, [28]).

We performed some experiments on different forms of mutation. Similarly as done by Pal [28] we used (1) 2-opt mutation or (2) series of 2-opt mutations until no such mutation can improve the tour. Using the genetic algorithm with the first mutation scheme it is relatively easy to bring the

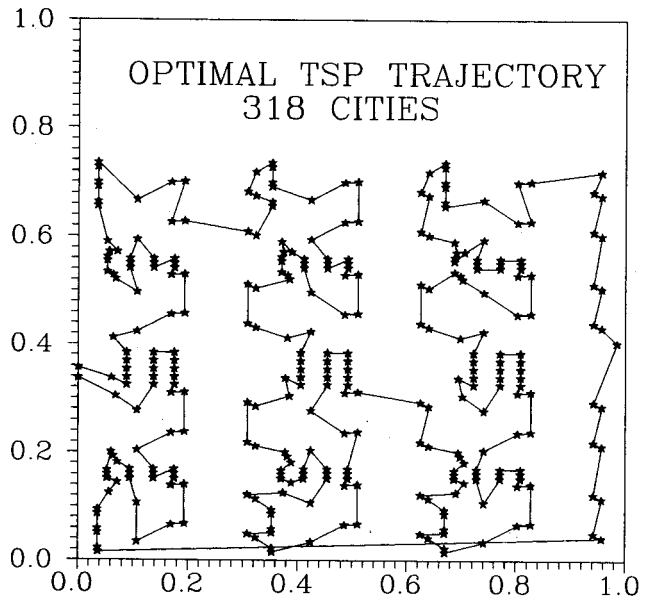


Fig. 2. Optimal trajectory for 318-cities TSP problem of Lin and Kernighan [17].

result to 43000 for the 318 city problem. The second mutation scheme, as already established by Pal [28] eventually brings the result to the exact length of 41345 (the library of TSP benchmarks [19]), see Fig. 2. The length achieved by Prinetto *et al.* was higher (although they also use 2-opt moves — they however, called these a hybrid: genetics algorithm plus heuristics rather than 2-opt mutations as Pal, [28]).

We have studied effect of the type (1) mutation scheme on speed of convergence implemented using 10000 individuals (with the mutation probability ranging from 0.1 to 0.9, which is very high). During the beginning stages of evolution, the crossover is much more effective than mutation. The 2-opt mutations were efficient only for tours not far from optimum.

2.4. Simulated annealing and tunneling

The most transparent presentation of simulated annealing (SA) in application to combinatorial optimization can be found in the recent edition of *Numerical Recipes* [1] (Chapter 10.9). We follow their presentation. The method uses analogy with thermodynamics of cooling. When the process is rapid a liquid metal is quenched in a meta-stable amorphous state. Slow cooling leads to perfect crystal, the lowest energy state. The expectation is that all processes of minimization follow a similar way. The standard

method assumes the Boltzmann probability distribution

$$\text{Prob}(E) = \exp\left(-\frac{E[s]}{k_B T}\right), \tag{6}$$

of states with different cost functions (energies). Following the Metropolis algorithm the probability of changing energy E_1 to E_2 is given by

$$p(\Delta E) = \exp\left[-\frac{(E_2 - E_1)}{k_B T}\right]. \tag{7}$$

If $E_2 > E_1$ a move is executed, otherwise a random number r is generated and if $r < p(\Delta E)$ there can still be a change of configuration. Since the system can go uphill one can escape from a local minimum. Implementing the Metropolis algorithm for a system different from thermodynamic requires three elements:

1. definition of a cost function equivalent to energy in physics, constraints usually are built in the definition of a configuration,
2. specification of possible changes of configuration,
3. control of annealing schedule.

For the TSP problem the configurations are fixed positions of cities on a plane (typically for testing purpose they are restricted to a square) labeled $1,\ldots,N$, together with the order of visiting each city represented by permutations of a sequence $1,\ldots,N$. The cost function is taken as the total length of the trip. The last two factors are very important. In statistical physics usually simple moves (such as flipping a spin) are considered. An analog of such procedure for the TSP would be the following. Select cities i,j not connected by the actual path. Change segments $i-1,i,i+1,\ldots,j-1,j,j+1$ into $i-1,j,i+1,\ldots,j-1,i,j+1$ (this is called a node insertion). As we will see one gets mediocre results using only this procedure. A change to a system using this procedure is quite abrupt compared to a change of a spin, which is a local change, or a change of the path trajectory in the path integral method. A much better strategy would be to execute a rearrangement that is most successful in a heuristic method. Implementation of *Numerical Recipes* consist of two moves: (a) a section of a path is removed then replaced with the same cities running in the opposite order (2-opt), (b) a section of a path is removed then replaced in between the two cities on another, randomly chosen part of a path (a part of full 3-opt, coming from a sequential 2-opts). The annealing schedule also effects the results (this actually is important in any method with a parameter controlling the convergence, *e.g.* using the Hopfield neural networks). Geman and Geman [30] have shown that for the Boltzmann Annealing the system can find a global

minimum (however in infinite time) if the annealing schedule is lowered at a rate

$$T(i) = \frac{T_0}{\log i}, \tag{8}$$

or slower. For the classical simulated annealing as a function of iteration i the temperature reads

$$T(i) = \frac{\alpha}{\log(1+\sigma i)}. \tag{9}$$

In the version called Fast Simulated Annealing [31], the cooling schedule is

$$T(i) = \frac{\gamma}{(1+\sigma i)}. \tag{10}$$

Typical parameters are $\alpha = 25, \gamma = 100, \sigma = 0.01$.

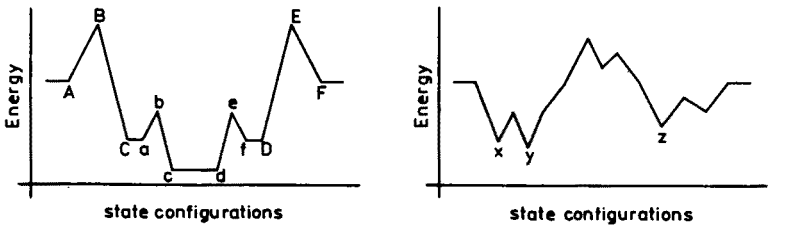


Fig. 3. Fractal and quasi-fractal energy landscapes.

The changing mechanism (type of moves) together with the annealing schedule determines an energy landscape. A recursive changing mechanisms can lead to a quasi-fractal energy landscape (see Fig. 3). It is conjectured that the concept of a quasi-fractal energy landscape can be formalized in ultra-metric space [4]. Lister made use of this concept for Ising model [32] and TSP [32, 33].

Another fruitful concept is to replace the Boltzmann distribution function for sampling with a function (multi-canonical simulated annealing) that reads

$$p(E) = \frac{\exp\left[-\frac{(\beta_i E + \alpha_i)}{k_B T}\right]}{Z}, \tag{11}$$

where

$$Z = \sum_E \exp\left[-\frac{(\beta_i E + \alpha_i)}{k_B T}\right]. \tag{12}$$

This energy probability distribution does not have clear physical interpretation. It is broader compared to canonical one and therefore two functions

in different local minima can overlap, and facilitate escape from a local minimum. Berg and Neuhaus [34] demonstrated the efficiency of the method to simulation of systems with first order transitions which led to rapid increase of convergence. Parameters β_i and α_i vary during iteration and are chosen empirically.

The idea of broadening the distribution (non-Boltzmann weight) is now universally accepted and was even earlier applied by Rujan [8] by the name of simulated tunneling where in contrast to simulated annealing the escape from a local minimum occurs by a transition under the barrier, and not over it due to thermal fluctuations. Rujan reported that with this method he achieved presumably the exact solution of length 7.485 on a 100 city problem with the 2-opt.

Similar broadening appears in the Constrained Global Optimization (CGO) method by Altschuler *et al.* [35]. The crucial part of their algorithm concerns determination whether a new value of a variable $x_i^{(n+1)}$ remains equal to the old value $x_i^{(n)}$ or not. The ratio of probabilities for this spin-flip like process is

$$\frac{p_{\text{old}}}{p_{\text{new}}} = \exp\left[-\frac{(g(x_i^n) - \alpha^n)}{k_{\text{B}}T}\right], \tag{13}$$

where $g(x)$ plays the role of a local cost function. We found, however, that CGO does not give better results than the SA for 2-opt and restricted set of 3-opt moves [36]. This is little surprising because the results for the office assignment problem were much in favour of CGO. These results raise several questions: (1) a given problem may have suitable heuristics, such as Lin, Kernighan for the TSP. Such heuristics may not work well for other problems, such as office assignment. (2) The lack of suitable heuristics increases attractiveness of general methods, such as simulated annealing or CGO.

The views of physicists and information scientists on SA differ widely. Physicists tend to think that the SA will be always better than heuristic approach. This might be true (but it has not been systematically studied) if heuristic moves contain no up-hill climbing mechanism. However, the statement is not generally true, because many best heuristic methods contain an up-hill climbing mechanism. In this respect, the moves in the work by Kirpatrick, Gelatt and Vecchi [3], which originated SA, for the 98 chip optimization must had probably been exceptionally well selected. Rather as verified by information scientists, the SA with a general set of moves will in general give inferior results compared to specialized heuristics. The better measure of the relative quality of SA would be to compare heuristics with SA with the same heuristic moves. For the 400 cities in the unit square Kirkpatrick [3] reported that the SA with 2-opt moves performs better than

2-opt and 3-opt (also the SA with the 2-opt moves is faster than 3-opt). It is not known how general is this result, and what is its dependence as a function of number of cities. Analogy with statistical mechanics implies that the performance of the SA would be relatively better compared with heuristics with an increasing number of cities. This may not hold for combinatorial problems which are not ergodic, particularly near a global minimum. Also it is not practical to verify performance of SA with the LK moves (that already use the up-hill mechanism) since it would be prohibitive for large systems.

An important recent development is the work by Lee and Choi [16] who applied multi-canonical SA for systems up to 40000 cities. Starting from 2-opt configurations they investigated the scaling relation of the tour length that for random distribution of cities behaves as $l(N) = \alpha\sqrt{N}$. For a 31623 city problem they found a tour with $\alpha(N) = 0.723$ slightly better than the LK heuristics. Although worse than the iterative LK heuristics giving $\alpha(N) = 0.716$ this illustrates the power of multi-canonical SA method.

The results of various methods are shown in Table I for the most thoroughly studied lin318 system. Simulated annealing with purely random move gives the average length around 58000. The result does not seem to be good. The reason is that random change of a city presents a tremendous change of a configuration. This is in contrast to simulations in physics, when a move of a spin causes a local change. The SA with powerful set of moves by Lin and Kernighan gives the average length around 43000.

TABLE I

Comparison of results for the lin318 city problem

Method	Tour length	average distance from the exact solution
Lin & Kernighan [17]	41883	1.3%
Hopfield neural network [21]	61337 – 64552	48.4 – 56.1%
Mean field [22]	55000 – 58000	33.0 – 40.3%
Simulated annealing with random moves	55000	33.0%
Simulated annealing + heuristic method [1]	43500	5.2%
Genetic algorithm	43000	4.0%
Genetic algorithm + heuristic method	41345	0.0%

3. Data clustering

Data clustering is a fundamental procedure in data processing, applicable to vector quantization [6], pattern recognition and self-organization. Examples include quantization of transmission signals or image compression, where Vector Quantization (VQ) is a basic compression technique at low bit-rate, below 1 bit/pixel (bpp) [6]. It consists in mapping of k-dimensional Euclidean space R^k into a finite subset Y of R^k. VQ consists of three stages: code book design, encoding and decoding. In this review we will concentrate on the encoding stage. The encoding procedure consists in finding for each input vector (otherwise called data or a training vector) $x = (x_1, x_2, \ldots, x_k)$ the closest code word (reference vector) $y_l = (y_1, y_2, \ldots, y_k)$ that minimizes the distortion $d^2(x, y_l) = \min d^2(x, y_j)$ for $j = 1, 2, \ldots, N$ (where N is a number of elements of a codebook) measured by the squared Euclidean distance

$$d^2(x, y_j) = \sum_{n=1}^{k} (x_n - y_{jn})^2 .$$

For the whole set of training vectors one minimizes the quantity

$$E = \sum_{i,j} A_{i,j} d(x_i, y_j) , \tag{14}$$

where $A_{i,j}$ is a probability of x_i to belong to cell C_j. One of the most important techniques is a k-means method (also called the Linde–Buzo–Gray (LBG) algorithm [38]). The method consists of two parts: *(i)* given reference vectors every datum is assigned to the nearest reference point (regions belonging to different reference vectors form a Voronoi partition) *(ii)* every reference point is moved to the gravity center of the data assigned to it. Iterating these two procedures leads to a local minimum. The Linde–Buzo–Gray algorithm [38] entails high computational complexity necessary for searching for the closest codeword. There are several methods that avoid the exhaustive search [39], and at the same time preserve the relatively high accuracy of the LBG full search algorithm.

There exists no complete comparison of general optimization methods with heuristic approaches for vector quantization. We would refer to two recent works together with our results. Ueda and Nakano [40] found that among the competitive learning methods, similar to Kohonen neural nets, the competitive and selective learning, CSL, in average is the best and gives by far smaller distortion than the k-means method. The quality of SA is controversial. Vaisey and Gersho [41] analyzed coding a set of 8192 training vectors with 256 16-dimensional code-vectors. The best result was achieved

with combination of SA with the LBG method, which however caused 25-fold increase in computation time compared to LBG. Surprisingly, $T_0 = 0$, which means accepting only down-hill moves in the cost function, made the results worse by a very small factor. Flanagan *et al.* [42] found for a set of image vectors similar to that used by Vaisey and Gersho [41], that they can achieve with the SA the results better and approximately taking similar computer time compared to the LBG. In their algorithm they moved training vectors between the Voronoi cells. Relatively very short time of their calculations was caused by a fast calculation of difference of distortion between consecutive steps (no information was given whether in LBG any fast search method was used). Shimoi and Lee [43] compared the entropic scheduling results (the best) with the k-means method results and simulated annealing (the worst).

We have undertaken our own SA study in view of conflicting information coming from the previous studies. We have tested the Adaptive Simulated Annealing (ASA) method [37] When simulated annealing is applied to optimization of functions the method consist of four relationships: (1) a function to optimize, $E(x)$ (2) $g(\Delta x)$ probability of density function in a search space of parameters, Gaussian in classical simulated annealing, N-dimensional Cauchy distribution for fast simulated annealing and

$$g_T(y) = \prod_{i=1}^{D} \frac{1}{2(|y_i| + T_i)\ln(1 + 1/T_i)}, \tag{15}$$

for ASA, where D is dimensionality of the space, (3) probability of accepting a new value, typically

$$h(x) = \frac{1}{1 + \exp\left(\frac{\Delta E(x)}{k_B T}\right)},$$

(4) the annealing temperature schedule. For ASA

$$T(k) = T_0 \exp(-ck^{1/D}), \tag{16}$$

where c is a constant.

Our results indicate that the simulated annealing method (being 1000 times slower than k-means method) for the Gaussian cluster fares worse than the k-means for one Gaussian cluster and little better for 3 separate Gaussian clusters of input data. These results are shown in Table II and calculation times were achieved on the IBM PowerPC C10 workstation. Clearly, the fact that heuristics (*i.e.* the LBG method) gives inferior results to competitive learning indicates that no algorithm comparable to the Lin–Kernighan one exists.

TABLE II

Comparison of results for LBG and ASA, for VQ; a Gaussian training vector set distribution, size of the codebook=4.

number of training vectors	LBG time [seconds]	LBG distortion	ASA time [seconds]	ASA distortion
300	0.03	0.7228	87.53	0.7185
1000	0.20	0.7266	276.44	0.7182
3000	0.58	0.6981	822.95	0.6988

4. Neural network implementation of a traffic control problem

Multistage interconnection networks have been widely used in ISDN and parallel computing. For the 3-stage Clos (n, m, r) networks various routing schemes have been reported in the literature for setting switches of a permutation network to realize a given permutation, or connection pattern from the inputs to outputs (see, Carpinelli and Oruc [44]). However, there is always need for general, although approximate methods of solution of routing and related optimization problems, and neural networks are quite used for this purpose [7]. The most extensive study of control problem was done by Brown and Liu [45]. They investigated performance of the Banyan controller. However, the optimal heuristic controller was found only for size of 8×8 problem. Hakim and Meadows [46] introduced the cost function to find optimal routing for a Benes switch.

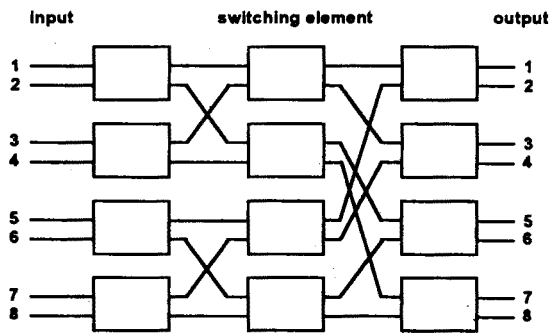


Fig. 4. The 8×8 reverse baseline network.

In the Funabiki, Takefuji, and Lee [47] work the Hopfield neural net [12] was used. They made a comparison of seven neural network models on traffic control problems in multistage interconnection networks. They used the

reverse baseline network. 8×8 version of it is shown in Fig. 4. The network is assumed to buffer incoming demands and be operated synchronously. However, contrary to more extensive Brown and Lee simulation only one time slot is considered. Transmission demands through the $n \times n$ network are represented by an $n \times n$ traffic matrix.

The cost function used by Funabiki *et al.* had the following form

$$E_{\mathrm{FTL}} = \frac{A}{2} \sum_{i=1}^{N} \left(\sum_{k=1}^{N} V_{ik} - 1 \right)^2 + \frac{A}{2} \sum_{j=1}^{N} \left(\sum_{k=1}^{N} V_{kj} - 1 \right)^2 + B \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{p \neq i}^{N} \sum_{q \neq j}^{N} s_{ijpq} V_{ij} V_{pq} . \tag{17}$$

This function represents objective of the problem including the constraints. The goal of the net is to minimize $E(V_1, V_2, \ldots, V_n)$ by solving the corresponding motion equation. The first (second) term is zero if there is no more than one 1 in each row (column). The third term imposes the constraints for adherence to the reverse baseline architecture, specified by s_{ijpq}.

The output $V_{ip} = 1$ of neuron of number i,p indicates whether the demand T_{ip} is selected. Their simulation showed that among the seven versions the hysteresis McCulloch–Pitts neuron model without the decay term and with two ad-hoc heuristics helping to escape from the local minima has the best performance. We have verified that using another cost function E_{CJ}:

$$E_{\mathrm{CJ}} = \frac{A}{2} \sum_{i,p \neq q} V_{ip} V_{iq} + \frac{B}{2} \sum_{i \neq j,p} V_{ip} V_{jp} - C \sum_{ip} V_{ip} + D \sum_{ip} \sum_{j \neq i} \sum_{q \neq p} s_{ijpq} V_{ip} V_{jq} + \frac{S}{2} \sum_{ip} V_{ip}^2 , \tag{18}$$

in which the first two terms in E_{FTL} are replaced by the constraints used by Mehta and Fulop [21] (which were the best for the Traveling Salesman Problem) does not improve the result. In conclusion, the Funabiki *et al.* approach is close to optimal as far as neural networks are concerned.

In order to assess a quality of performance of the Hopfield neural network it is necessary to compare it to any other method, which has not been routinely done in the traffic control field. To this end we proposed a very simple heuristic algorithm [48] which for the sake of brevity will not be discussed here. The results of the neural network approach and our heuristic algorithm for the traffic control problem are compared in Table III.

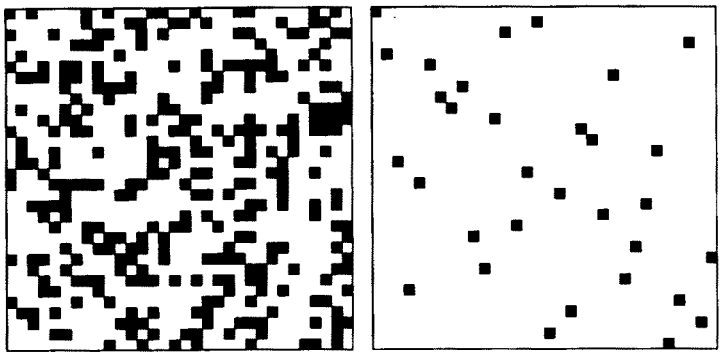
TABLE III

Comparison of performance of Funabiki *et al.* neural network method [47] with a simple heuristic algorithm [48].

	Density	frequency of convergence to global minimum	frequency of convergence to local minimum	quality of a local minimum
Our	10%	0%	100%	17.81
method [48]	20%	5%	100%	28.36
$N = 32$	30%	56%	100%	31.37
	40%	97%	100%	31.97
	50%	100%	100%	32.00
	80%	100%	100%	32.00
Neural				
Network	50%	87%	100%	31.80
Approach	80%	100%	100%	32.00
Funabiki *et al.* $N = 32$				
Our	10%	0%	100%	47.58
method	20%	16%	100%	62.25
$N = 64$	30%	92%	100%	63.92
	40%	99%	100%	63.99
	50%	100%	100%	64.00
	80%	100%	100%	64.00
Our	10%	0%	100%	113.6
method	20%	73%	100%	127.7
$N = 128$	30%	100%	100%	128.0
	40%	100%	100%	128.0
	50%	100%	100%	128.0
	80%	100%	100%	128.0
Our	10%	0%	100%	250.5
method	20%	100%	100%	256.0
$N = 256$	30%	100%	100%	256.0
	40%	100%	100%	256.0
	50%	100%	100%	256.0
	80%	100%	100%	256.0

They are given as a function of traffic density, *i.e.* percentage of 1's in random traffic matrices. The results from the heuristic algorithm approach represent an average of 100 traffic matrices for $N < 256$, and an average of 10 traffic matrices for $N = 256$. It is seen that already for $N = 32$ the best version of the neural network is not capable to find a global minimum for the traffic density 50%. The heuristic algorithm finds a global minimum

(optimal routing) for 50% density up to $N = 256$, and actually is very close to optimum even for traffic density 20%.



(a) (b)

Fig. 5. (a) A 32×32 traffic matrix with 30% density (b) a global optimal solution. Black entries correspond to $T_{ip} = 1$, while white entries correspond to $T_{ip} = 0$.

We show an example of traffic matrix for $N = 32$ (Fig. 5) together with optimal solutions for these cases.

5. Conclusions

This review presents a comparison of performance of several methods for various combinatorial problems. Generally the Hopfield type neural networks give the worst performance. Their biggest drawback is occurrence of illegal solutions violating the constraints defining a model. The competitive learning networks perform much better.

Random move simulated annealing gives poor quality of solutions and is slow. The multi-canonical SA seems to be the best in this class and deserves testing in application to wide spectrum of optimization problems. No method (genetic algorithms being closest) can outperform the best heuristic algorithm (like the LK one for the Traveling Salesman Problem) when it exists. The general purpose algorithms achieve the best results when they incorporate heuristic moves. Understanding why certain moves perform better (to what extent they probe longer-ranged correlations) is an open question. We also conjecture that physics could gain from experience of information science with heuristic algorithms by incorporating some of their features into simulations of purely physical systems. Recently two new algorithms (particularly, the first one) Tabu Search [49] and Ant Algorithm [50] are gaining popularity. In Tabu Search there exists a builtin mechanism that forbids returning to the same feasible solution. The Tabu

Search for TSP becomes main competitor of the Lin-Kerninghan strategy in the class of heuristic algorithms. The Ant Algorithm mimics behavior of ant and termite colonies to find the shortest path between their living sites and feeding sources. This algorithm is well suited for calculations using distributed systems.

The emergence of these algorithms indicates potential for new promising heuristic strategies.

This work has been partially supported by the US-Polish Sklodowska-Curie Joint Fund II MEN/NSF/92-116. The authors are grateful to Drs. P. Rujan (for discussions on the TSP), R. Lister for sending his Ph.D. thesis with comparison of the simulated annealing and the heuristic methods and Prof. A. Jajszczyk for introducing them to traffic control problem in multistage interconnection networks, and numerous discussions and comments.

REFERENCES

[1] W.H. Press, S.A. Teukolsky, W.T. Vetterling, P.B. Flannery, *Numerical Recipes in C*, Cambridge Univ. Press, 1992.
[2] S. Kirkpatrick, *J. Stat. Phys.* **34**, 975 (1984).
[3] S. Kirkpatrick, C.D. Jr. Gelatt., M.P. Vecchi, *Science* **220**, 671 (1983).
[4] S. Kirkpatrick, G. Toulouse, *J. Phys.* **46**, 1277 (1985).
[5] E.L. Lawler, J.K. Lenstra, A.G.H. Rinnoy, D.B. Shmoys, *The Traveling Salesman Problem*, J. Wiley, New York 1985.
[6] A. Gersho, R.M. Gray, *Vector Quantization and Signal Compression*, Boston, Kluwer 1992.
[7] T.X. Brown, *IEEE Commun. Magazine* **27**, 72 (1989).
[8] P. Rujan, *Z. Phys.* **B73**, 391 (1988).
[9] R. Lister, On Making the Right Moves: Neural Networks, Gradient Descent, and Simulated Annealing, Ph. D. thesis (unpublished), University of Sydney, Feb. 1992.
[10] M.A. Styblinski, T.S. Tang, *Neural Networks*, **3**, 467 (1990).
[11] M.W. Padberg, G. Rinaldi, *SIAM Rev.* **33**, 1 (1991).
[12] J.J. Hopfield, D.W. Tank, *Biol. Cybern.* **52**, 141 (1985).
[13] T. Kohonen, *Self-Organization and Associative Memory*, Springer-Verlag, Berlin 1984.
[14] B. Angeniol, G. Vaubois, J. Texier, *Neural Networks* **1**, 289 (1988).
[15] R. Durbin, D. Willshaw, *Nature* **326**, 681 (1987).
[16] J. Lee, M.Y.Choi, to be published (1995).
[17] S. Lin, B.W. Kernighan, *Operations Research* **21**, 498 (1973).
[18] H. Crowder, M.W. Padberg, *Management Sci.* **26**, 496 (1980).
[19] G. Reinelt, TSPLIB 1.2, Institut für Mathematik, Universitët Augsburg, preprint Dec. 1990.

[20] M. Jünger, G. Reinelt, G. Rinaldi, in *Networks, Handbooks in Operation Research and Management Science Series* eds. M.O Ball *et al.*, 1995, to be published.
[21] S. Mehta, L. Fulop, *Neural Networks* **6**, 869 (1993).
[22] C. Peterson, B. Södeberg, *Int. J. Neural Systems* **1**, 3 (1989).
[23] C. Peterson, *Neural Comp.* **2**, 261 (1990).
[24] L. Ciepliński, C.Jędrzejek, *Appl. Math. Comp. Sci.* **4**, 423 (1994).
[25] J. Grefenstette, R. Gopal, B. Rosmaita, D. Van Gucht, Proc. First Int. Conference On Genetic Algorithms, Pittsburgh, PA (USA), 160-168 (1985).
[26] H. Muhlenbein, M. Schomisch, J. Born, *Parallel Computing* **17**, 619 (1991).
[27] F.Q. Bac, V.L. Perov, *Biol. Cybern.* **69**, 229 (1993).
[28] K.F. Pal, *Biol. Cybern.* **69**, 539 (1993).
[29] P. Prinetto, M.Rebaudengo, M.S. Reorda, Proc. Int. Conf. on Artificial Neural Nets and Genetic Algorithms, Insbruck, Austria, Springer-Verlag, 1993, p.559.
[30] S. Geman, D. Geman, *IEEE Trans. on Pattern Analysis and Machine Inteligence*, **6**, 721 (1984).
[31] H. Szu, Proceedings of *IEEE*, **75**, 1538 (1987).
[32] R. Lister, *IEEE Int. Conf. on Neural Networks*, San Francisco, Vol. I, 1993, p.257.
[33] R. Lister, Int. Joint Conf. on Neural Networks, Vol. I, 1990, p.424.
[34] B.A. Berg, T. Neuhaus, *Phys. Lett.* **B267**, 249 (1991); *Phys. Rev. Lett.* **68**, 9 (1992).
[35] E.L. Altschuler, T.J. Williams, E.R. Ratner, F.Dowla, F. Wooten, *Phys. Rev. Lett.*, **72**, 2671 (1994).
[36] D. Kudzia, L. Ciepliński, C.Jedrzejek, to be published.
[37] A.L. Ingber, *J. Mathl. Comput. Modelling* **18**, 29 (1993).
[38] Y. Linde, A. Buzo, R.M. Gray, *IEEE Trans. Commun.* **C-28**, 84 (1980).
[39] C.-H. Lee, L.-H. Chen, *IEE Proc.-Vis. Image Signal Process* **141**, 143 (1994).
[40] N. Ueda, R. Nakano, *Neural Networks* **7**, 1211 (1994).
[41] J. Vaisey, A. Gersho, in Proceedings of ICASSP, 1988, p.1176.
[42] J.K. Flanagan, D.R. Morrell, R.L. Frost, C.J. Read, B.E. Nelson, in Proceedings of ICASSP, 1989, p.1759.
[43] S. Shimoji, S. Lee, to be published (1995).
[44] J.D. Carpinelli, A.Y. Oruc, *IEEE Trans. Commun.* **41**, 1245 (1993).
[45] T.X. Brown, K.H. Liu, *IEEE J. Select. Areas Commun.* **8**, 1428 (1990).
[46] N.Z. Hakim, H.E. Meadows, *Proceedings IEEE INFOCOM '90*, **2**, 397 (1990).
[47] N. Funabiki, Y. Takefuji, K.Chun-Lee, *IEEE Trans. Computers*, **42**, 497 (1993).
[48] L. Ciepliński, C.Jedrzejek, to be published.
[49] F. Glover, **1**, 190 (1990); *ORSA J. Comput.* **2**, 4 (1990).
[50] A. Colorni, M. Dorigo, V. Maniezzo, Proceedings of the Parallel Problem Solving from Nature Conference, Brussels, Belgium, Elsevier Publishing, 1992, p.509.