January 1990

LU TP 90-2

Parallel Distributed Approaches to Combinatorial Optimization - Benchmark Studies on Traveling Salesman Problem

Carsten Peterson[1]

Department of Theoretical Physics, University of Lund
Sölvegatan 14A, S-22362 Lund, Sweden

Neural Computation **2**, 261-269 (1990)

Abstract:

We present and summarize the results from 50-, 100- and 200-city TSP benchmarks presented at the 1989 NIPS post-conference workshop using neural network, elastic net, genetic algorithm and simulated annealing approaches. These results are also compared with a state-of-the-art hybrid approach consisting of greedy solution, simulated annealing, and exhaustive search.

[1] carsten@thep.lu.se

Background

Using neural networks to find approximate solutions to difficult optimization problems is a very attractive prospect. In the original paper [1] 10- and 30-city traveling salesman problems (TSP) were studied with very good results for the N=10 case. For N=30 the authors report on difficulties in finding optimal parameters. In ref. [2] further studies of the Tank-Hopfield approach were made with respect to refinements and extension to larger problem sizes. The authors of ref. [2] find the results discouraging. These and other similar findings have created a negative opinion about the entire concept of using neural network algorithms for optimization problems in the community.

Recently a novel scheme for mapping optimization problems onto neural networks was developed [3]. The key new ingredient in this method is the reduction of solution space by one dimension by using multi-state neurons (**Potts** spin [4]), thereby avoiding the destructive redundancy that plagues the approach of the original work by Tank and Hopfield [1]. The idea of using Potts glass for optimization problems was first introduced by Kanter and Sompolinsky [5]. This encoding was also employed in [6]. Very encouraging results were found when exploring this technique numerically [3].

An alternative approach to solve TSP in brain-style computing was developed in refs. [7][8], where a feature map algorithm is used. Basically, an elastic "rubber band" is allowed to expand to touch all cities. The dynamical variables are the coordinates on the band, which vary with a gradient descent prescription on a cleverly chosen energy function. It has recently been demonstrated that there is a strong correspondance between this **elastic net** algorithm and the Potts approach [9][10][11].

Parallel to these developments **genetic algorithms** have been developed for solving these kind of problems [12][13] with extremely high quality results.

Given the above mentioned scepticism towards the neural network approach and the relatively unknown success of the genetic approach we found it worthwhile to test these three different parallel distributed approaches on a common set of problems and compare the results with "standard" simulated annealing. The simulations of the different algorithms were done completely independently at different geographic locations and presented at the 1989 NIPS post-conference workshop (Keystone, Colorado). In order to further increase the value of this mini-report we have also included comparisons with a hybrid approach consisting of greedy solutions, exhaustive search and simulated annealing [20].

The testbeds consisted of 50-, 100- and 200-city TSP with randomly chosen city coordinates within a unit square. All approaches used an identical set of such city coordinates. The reason for chosing TSP is its widely acceptance as a NP-complete benchmark problem. The problem sizes were selected to be large enough to challenge the algorithms and at the same time feasable with limited CPU availability. Since the neural network approaches are known to have a harder time with random (due to the mean field theory averaging involved) than structured problems [14] we chose the former.

The Algorithms

Before comparing and discussing the results we briefly list the key ingredients and parameter choices for each of the algorithms.

The Potts Neural Network [3]

This algorithm is based on an energy function similar to the one used in the original work by Tank and Hopfield [1]

$$E = \sum_{ij} D_{ij} \sum_{a} S_{ia} S_{j(a+1)} - \frac{\beta}{2} \sum_{i} \sum_{a} S_{ia}^2 + \frac{\alpha}{2} \sum_{a} (\sum_{i} S_{ia})^2 \tag{1}$$

In eq. (1) the first term miminizes the tour length (D_{ij} is the intercity distance matrix), and the second and third terms ensure that each city is visited exactly once. A major novel property is that the condition

$$\sum_{a} S_{ia} = 1 \tag{2}$$

is always satisfied; the dynamics is confined to a hyperplane rather than a hypercube. Consequently the corresponding mean field equations read

$$V_{ia} = \frac{e^{U_{ia}}}{\sum_b e^{U_{ib}}} \tag{3}$$

where $V_{ia} =< S_{ia} >_T$ and the local fields U_{ia} are given by

$$U_{ia} = -\frac{1}{T} \frac{\partial E}{\partial V_{ia}} \tag{4}$$

The mean field equations (eq. (3)) are minimizing the free energy ($F = E - TS$) corresponding to E in eq. (1). A crucial parameter when solving eqs. (3,4) is the temperature T. It should be chosen in the vicinity of the critical temperature T_c. In

N	M	α	β	K_0	N_{iter}
50	125	0.2	2.0	0.29	300
100	250	0.2	2.0	0.26	300
200	500	0.2	4.0	0.27	182

Table 1: Parameters used for the Elastic Net algorthm. The parameters α and β are chosen to satisfy conditions for valid tours [8].

ref. [3] a method for estimating T_c in advance by estimating the eigenvalue distibution of the linearized version of eq. (3). This turns out to be very important for obtaining good solutions.

For the details of annealing schedule, choice of α, β etc. used in this benchmark study we refer to the "black box" prescription in Sect. 7 of ref. [3].

The Elastic Net [7]

This approach is more geometrical. It is a mapping from a plane to a circle such that each city on the plane is mapped onto a point on the circle (path). The N city coordinates are denoted $\vec{x}_i$. Points on the path are denoted $\vec{y}_a$, where $a = 1, ..., M$. Note that M can in principle be larger than N. The algorithm works as follows:

Start with a small radius circle containing the M $\vec{y}_a$-coordinates with an origin slightly displaced from the center of gravity for the N cities. Let the $\vec{y}$-coordinates be the dynamical variables and change them such that the energy

$$E = -\alpha K \sum_i^N \log \sum_a^M e^{|\vec{x}_i - \vec{y}_a|^2/2K^2} + \beta \sum_a^M |\vec{y}_{a+1} - \vec{y}_a|^2 \tag{5}$$

is minimized. Gradient descent on eq. (5) causes the initial circle to expand in order to minimize the distances between $\vec{y}$ and $\vec{x}$ coordinates in the first term of eq. (5) at the same time as the total length is minimized by the second term. Good numerical results were obtained with this method with $M > N$ [7]. The parameter K in eq. (5) has the role of a temperature and as in the case of the neural network approach above a critical value K_0 can be computed from a linear expansion [8]. The values of the parameters used in this benchmark [15] can be found in Table 1.

This algorithm is closely related to the Potts neural network [9][10][11]. Loosely speaking this connection goes as follows. Since the mean field variables V_{ia} are probabilities (cf. eq. (2)) the average distances between tour positions $d_{ab} = |\vec{y}_a - \vec{y}_b|$

and average distances between and cities and tour positions $\tilde{d}_{ia} = |\vec{x}_i - \vec{y}_a|$ can be expressed in terms of the distance matrix D_{ij}. The second term in eq. (5) can then be identified with the tour length term in eq. (1) if the metric is choosen to be D_{ij}^2 rather than D_{ij}. The first term in eq. (5) corresponds to the entropy of the Potts system (eq. (1)); gradient descent on eq. (5) corresponds to minimizing the free energy of the Potts system, which is exactly what the MFT equations are doing.

There is a difference between the two approaches, which has consequences on the simulation level. Each decision element S_{ia} in the Potts neural network approach consist of two binary variables, which in the mean field theory treatment becomes two analog variables; N cities requires N^2 analog variables. In the elastic net case N cities only requires $2M (M > N)$ analog variables; it is a more ecomical way of representing the problem.

The Genetic Algorithm [13] [16]

For details we refer the reader to refs. [13] [16]. Here we briefly list the main steps and the parameters used.

1. Give the problem to M individuals.

2. Let each individual compute a local minimum (2-quick[2]).

3. Let each individual choose partner for mating. In contrast to earlier genetic algorithms [12] global ranking of all individuals is not used. Rather local neighborhoods of size $D - 1$ were used in which the selection is done with weights $(r_1, r_2, ..., r_D)$. The global best gets weight r_1 and and the remainding local neighbors get $r_2, ..., r_D$ respectively.

4. Crossover and Mutation. A random string of "genes" is copied from the parent to the offspring. The string size is randomly chosen in the interval $[c_1, c_2]$. Mutation rate=m.

5. If not converged return to pt. 2.

The parameters used for the benchmarks [17] are shown in Table 2.

[2] This is the 2-opt of ref. [18] with no checkout.

N	M	D	$r_1 : r_2 : .. : r_D$	$[c_1, c_2]$	m	N_{gen}
50	64	8	0.25:0.20:0.15:0.10:0.10:0.10:0.05	[N/4,N/2]	0.01	30
100						23
200						562

Table 2: Parameters used for the genetic algorithm. The choice of $M = 64$ was motivated by the available 64 T800 transputer configuration of ref. [17].

Simulated Annealing [19]

The parameters of this algorithm are the initial temperature T_0, and the annealing schedule, which determines the next value of T and the length of time L spent at each T. The annealing schedule used is very generous (see figure 1) and is based upon a requirement that the temperature be high enough such that the variance of the energy is less than T:

$$\left(\langle E^2\rangle - \langle E\rangle^2\right)/T \ll 1. \tag{6}$$

1. $T_0 = 10$; $L_0 = N$.
2. Until variance of cost function < 0.05, update according to:
   $T = T/0.8$; $L = N$ (heating up).
3. While percentage of accepted moves $> 50\%$, update according to:
   $T = 0.95 \times T$; $L = N$ (cooling).
4. Until number of uphill moves $= 0$, update according to:
   $T = 0.95 \times T$; $L = 16N$ (slow cooling).

Figure 1: Annealing schedule.

A Hybrid Approach [20][3]

The above parallel distributed and simulated annealing approaches are clean in the sense that each of them relies on a single algorithm with no optimization of initial states etc.. It would not be surprising that if one wants to push the quality to the limits a hybrid approach would be most rewarding. Indeed this turns out to be the case for a combination of greedy, simulated annealing and exhaustive search approaches [20]. The procedure is as follows:

1. A greedy tour is obtained by starting at city #1, then proceeding to the nearest remaining city, then the nearest remaining city, and so on until all cities have been visited once.
2. Simulated annealing, using 2-bond and restricted 3-bond rearrangements as "moves", is used to equilibrate the tours at a temperature chosen to yield approximately the same length as found by the greedy step 1. Subsequent annealing lower the temperature to shrink the tours until accepted moves are scarce.
3. Exhaustive search is performed on the result of step 2, using 2-bond rearrangements until no more improvements are found, then restricted 3-bond rearrangements. If a 3-bond improvemnt is found, the exhaustive search resumes with 2-bond moves, halting when no improvements are found using either type of move.

Results

In Table 3 we compare the averaged performance of the neural network (5 trials), elastic net (1 trial), genetic algorithm (1 trial), simulated annealing (5 trials) and hybrid (5 trials) algorithms. For comparison we have also included the results from random distributions based on 1000 trials. Results of the greedy solutions (step 1 in the hybrid approach) are approximately equal to the figures reported for NN and SA in Table 3.

Summary and Comments

The overall performance of the three different parallel distributed approaches is impressive.

[3] The author is responsible for denoting this method "hybrid".

N	NN	EN	GA	SA	HA	RD
50	6.61	5.62	5.58	6.80		26.95
100	8.58	7.69	7.43	8.68	7.48	52.57
200	12.66	11.14	10.49	12.79	10.53	106.42

Table 3: Performance averages (tour lengths) of the neural network (NN), elastic net (EN), genetic algorithm (GA), simulated annealing (SA), hybrid approach (HA) and random distribution (RD).

- They are all better or equal to "standard" simulated annealing. These methods are here to stay! The genetic algorithm is the winner closely followed by the hybrid approach by [20][21] and the elastic net method.

- Even though the neural network approach does not fully match the performance of the other two the results invalidate the common saying: "Neural networks optimization algorithms do not work for $N > 30$ problems". This algorithm has also been tested on larger problem sizes [22] than presented in this report with no sign of quality deterioration[4].

- It is somewhat surprising that the performance of the neural network algorithm is of less quality than that of the elastic net given the close connection discussed above. We believe this is due to the fact that the former is more sensitive to the position of T_c. Indeed, subsequent modifications of the annealing procedure and choice of T_c [22] have lead to better results[4]. Also, the performance of the NN algorithm can be substantially improved when starting out from a greedy solution, heating the system up and letting it relax with the MFT equations[4].

- Another important issue is computing time. It splits up into two parts: number of operations (in serial execution) per iteration for a given problem size N and the number of iterations needed for convergence. In table 4. we compare these numbers for the different algorithms. The convergence times in this table are all empirical. The numbers in Table 4 have limited value since the real strength in the distributed parallel approaches is their inherent parallelism.

Acknowledgements

I would like to thank Richard Durbin, Alan Yuille, Heinz Mühlenbein and Scott Kirkpatrick for providing simulation results from the elastic net [15], genetic algorithm

[4] In order to strictly stick to the NIPS presentations we did not to include these extensions and improvements in this report.

	no. of operations	no. of iterations	total
NN	$\sim N^3$	const.	$\sim N^3$
EN	$\sim N$	const.	$\sim N^2$
GA	$\sim MN$	$\sim N$ (?)	$\sim MN^2$ (?)
HA			$\sim N^2$

Table 4: Time consumption for the different algorithms when executed serially.

[17] and a hybrid approach [21] for these comparisons. Also the stimulating atmosphere provided by the organizers of the 1989 NIPS post-conference workshop is very much appreciated.

References

[1] J.J.Hopfield and D.W.Tank, "Neural Computation of Decisions in Optimization Problems", *Biological Cybernetics* **52**, 141 (1985).

[2] G.V.Wilson and G.S. Pawley, "On the Stability of the Travelling Salesman Problem Algorithm of Hopfield and Tank", *Bilogical Cybernetics* **58** 63 (1988).

[3] C. Peterson and B. Söderberg, "A New Method for Mapping Optimization Problems onto Neural Networks", *International Journal of Neural Systems* **1**, 3 (1989).

[4] F.Y. Wu, "The Potts Model", *Review of Modern Physics* **54**, 235 (1983).

[5] I. Kanter and H. Sompolinsky, "Graph Optimization Problems and the Potts Glass", *Journal of Physics* **A 20**, L673 (1987).

[6] D.E.Van den Bout and T.K. Miller III, "A Traveling Salesman Objective Function that Works", *Proceedings of the IEEE International Conference on Neural Networks*, pp. 299 (1988).

[7] R.Durbin and G.Willshaw, "An Analogue Approach to the Travelling Salesman Problem Using An Elastic Net Method", *Nature* **326**, 689 (1987).

[8] R.Durbin, R.Szeliski and A. Yuille, "An Analysis of the Elastic Net Approach to the Traveling Salesman Problem", *Neural Computation* **1**, 348 (1989).

[9] P.D. Simic, "Statistical Mechanics as the Underlying Theory of "Elastic" and "Neural" Optimizations", CALT-68-1556 & C3P-787 (to appear in *Network: Computation in Neural Systems*).

[10] A. Yuille, "Deformable Templates, Statistical Physics and Matching Problems: Harvard Robotics Laboratory Technical Report No, 89-12.

[11] C. Peterson and B. Söderberg, "The Elastic Net as a Mean Field Theory Neural Network", LU TP 90-18.

[12] J.H. Holland, *Adaption in Natural and Artificial Systems* (University of Michigan Press, Ann Arbor, 1975).

[13] H. Mühlenbein, M. Gorges-Schleuter and O. Krämer, "Evolution Algorithms in Combinatorial Optimization", *Parallel Computing* **7**, 65 (1988);

[14] C. Peterson and J.R. Anderson, "Applicability of Mean Field Theory Neural Network Methods to Graph Partitioning", MCC-ACA-ST-064-88.

[15] R. Durbin and A. Yuille, Elastic net simulations results provided for this benchmark, private communication.

[16] M. Gorges-Schleuter, "ASPARAGOS - An Asynchronous Parallel Genetic Optimization Strategy", *Proceedings of the Third International Conference on Genetic Algorithms*, pp. 422 , D. Schaffer (Ed.) Morgan Kaufmann Publ. San Mateo (1989).

[17] H. Mühlenbein, Genetic algorithm simulations results provided for this benchmark, private communication.

[18] S. Lin, "Computer Solutions of the Traveling Salesman Problem", *Bell Systems Techn. J* **44**, 2245 (1965).

[19] S.Kirkpatrick, C.D.Gelatt and M.P.Vecchi, "Optimization by Simulated Annealing", *Science* **220**, 671 (1983).

[20] S.Kirkpatrick, G.Toulouse, "Configuration Space Analysis of the Travelling Salesman Problem', *Journal de Physique* **46**, 1277 (1985).

[21] S.Kirkpatrick, Hybrid approach simulation results provided for this benchmark, private communication.

[22] C. Peterson, unpublished.