Biol. Cybern. 61, 11–19 (1989)

Biological Cybernetics

A Neural Network Algorithm for the Multiple Traveling Salesmen Problem

E. Wacholder, J. Han, and R. C. Mann

Oak Ridge National Laboratory, Building 6025, MS-6364, P.O. Box 2008, Oak Ridge, TN 37831-6364, USA

Abstract. We developed an efficient neural network algorithm for solving the Multiple Traveling Salesmen Problem (MTSP). A new transformation of the N-city M-salesmen MTSP to the standard Traveling Salesmen Problem (TSP) is introduced. The transformed problem is represented by an expanded version of Hopfield-Tank's neuromorphic city-position map with $(N+M-1)$-cities and a single fictitious salesmen. The dynamic model associated with the problem is based on the Basic Differential Multiplier Method (BDMM) [26] which evaluates Lagrange multipliers simultaneously with the problem's state variables. The algorithm was successfully tested on many problems with up to 30 cities and five salesmen. In all test cases, the algorithm always converged to valid solutions. The great advantage of this kind of algorithm is that it can provide solutions to complex decision making problems directly by solving a system of ordinary differential equations. No learning steps, logical if statements or adjusting of parameters are required during the computation. The algorithm can therefore be implemented in hardware to solve complex constraint satisfaction problems such as the MTSP at the speed of analog silicon VLSI devices or possibly future optical neural computers.

1 Introduction

The MTSP is an extension of the well-known TSP. It is a prototype of difficult combinatorial optimization problems, and has a variety of important applications, mostly in the areas of routing and scheduling problems, for example, the school bus routing problems (Angel et al. [1], Orloff [2], Christofides and Eilon [3], topological design of computer networks (Gavish [4, 5] and the dia-a-ride problem (Gavish and Srikanth [6]). Finding an optimal solution method for the MTSP is of great importance in many areas of science and engineering; for example, computer systems, robotics, operation research, economics, management, physics, communication network systems, military applications etc.

While the TSP is considered to be one of the standard problems in the operation research/management science literature, the MTSP has not yet received such wide attention. Both problems are classified as NP-complete (Garey and Johnson [29]). The amount of computation time required to solve such problems increases exponentially with the size of the problem.

Recent advances in the study of artificial neural networks have resulted, among others, in the algorithm of Hopfield and Tank [7] who showed how decision problems like the TSP, for instance, may be solved by neuron-like computational networks. Their work has attracted the attention of many researchers to the field of neural networks [7–14], however, no significant modifications to this algorithm have been reported since then. In the area of hardware implementation, major achievements have resulted from the efforts of Hecht-Nielsen, Mead, and others [15–17].

Neural network algorithms for computing solutions to optimization problems are basically variations of either the simulated annealing method [8–24] or the deterministic differential gradient descent approach as suggested by Hopfield and Tank [7]. Recently Durbin and Willshaw [25] have shown that their approach to the TSP using an elastic net method is very accurate. These authors claim that their algorithm is naturally extendable to a large class of optimization problems.

In this article we extend the neural network deterministic differential gradient descent approach of Hopfield and Tank [7] to solve the MTSP. We illustrate our method by applying it to the basic MTSP with constant M. This represents an optimization problem with only equality constraints.

We show that the MTSP can be transformed into an $(N+M-1)$-city TSP. The transformed problem is represented in a neural network, which is an expanded version of the two-dimensional city-position map suggested by Hopfield and Tank [7], that contains an additional $(M-1)$ duplicates of the base city. A corresponding expansion of the distance matrix is required to obtain a well-posed problem.

Hopfield and Tank's method monotonically decreases the value of an energy function which characterizes the TSP. By beginning the iteration at an arbitrary point in state space and going in the direction of steepest descent, the method can be shown to converge to valid tours, however, it can not be guaranteed to be stable and to converge to global minima. To solve this problem we have exploited the Basic Differential Method of Multipliers (BDMM) [25] for constrained optimization. This algorithm can be shown to satisfy the problem's constraints exactly, while leading to stable solutions.

The BDMM is a composition of the differential gradient descent algorithm and the method of Lagrange multipliers [25]. This is accomplished by adding ODEs to solve for the Lagrange multipliers simultaneously with the calculation of the problem's state variables.

The great advantage of this algorithm is that it can be implemented in hardware. Solutions to complicated decision making problems are obtained by solving a system of ODEs in time.

The problem is presented in Sect. 2. The algorithm is analyzed in Sect. 3, and simulation and discussion results are presented in Sect. 4. In Sect. 5 we summarize our findings.

2 Statement of the Problem and Representation Scheme

The MTSP we are going to consider in this paper is the following:

Given a set of N cities, find a set of closed routes for M salesmen starting from, and ending at, a given fixed city (base city), so that the total distance covered is minimized. The minimization is subject to the constraints that each city (apart from the base city) is visited exactly once by exactly one of the M salesmen and every salesman should visit at least r customer cities.

If $M=1$ the problem reduces to the standard TSP with a fixed base city.

To solve the problem using ANN (Artificial Neural Network) system, we need to map the problem onto a suitable network. This also requires a representation scheme in which a valid tour is specified by the output states of the neurons.

Hopfield and Tank [7] suggested an $N \times N$ permutation matrix as a representation scheme for the standard TSP of N cities. Their network consists of N^2 neurons, and outputs from this net are used to form an $N \times N$ matrix, which we will call a city-position map. Each row in this map corresponds to a particular city, while each column corresponds to a particular position in a tour.

Our approach to the MTSP is based on a transformation of the problem to an expanded form of the standard TSP. Suppose M salesmen, a set of cities $\{1, 2, \ldots, N\}$, and a matrix of distances between cities (d_{ij}) are given. The base city number is chosen to be 1 without loss of generality. To construct a city-position map, $(M-1)$ fictitious cities numbered $N+1$, $N+2, \ldots, N+M-1$ are added at the same location as the base city. Consider a tour of one salesman around these $(N+M-1)$ cities with the following restrictions:

R1. Start from the base city, visit all cities exactly once, and return to the base city.

R2. A valid tour consists of exactly M nontrivial subtours[1] that have only the base city in common.

Whenever a fictitious city appears in the middle of a tour, a salesman returns to the base city since the locations of fictitious cities coincide with the base city. Hence M closed loops are formed, and each closed loop will be assigned as a tour to each of M salesmen in the MTSP. The condition of nontriviality is necessary to make each salesman visit at least one customer city.

Consider, for instance, the case of $N=8$ cities and $M=3$ salesmen. We observe that (see Fig. 1):

i) The sequences 1 5 2 9 7 3 4 10 6 8 1 and 1 5 2 10 7 3 4 9 6 8 1 in Fig. 1 both represent valid tours. In general, we have $(M-1)$ degenerate tours in the expanded case for one valid MTSP tour.

ii) Since the fictitious cities are located at the base city, the distances covered by taking any one of the degenerate tours are identical, and the corresponding tour in the MTSP covers the same distance.

Conditions (R1) and (R2) suggest a $(N+M-1) \times (N+M-1)$ city-position permutation matrix with the following additional properties:

P1. A "1" should appear at the $(1, 1)$ position. This is the constraint that the base city is selected to be the first and last city to be visited by the fictitious single salesman.

P2. The $(i, 2)-(i, s+1)$ and

$$(i, N+M-s)-(i, N+M-1)$$

elements for $s=1, 2, \ldots, r$ in the last $(M-1)$ rows (i.e., $i=N+1, \ldots, N+M-1$), should be "0". This ensures

---

[1] A nontrivial subtour is one which consists of at least one city other than the base city

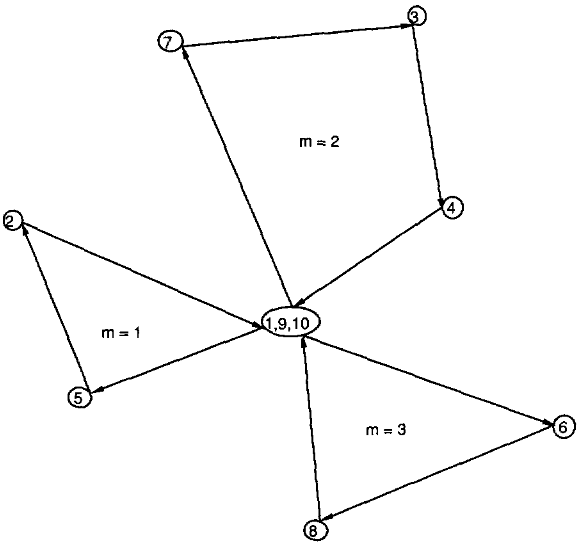


Fig. 1. A valid tour of 3 salesmen to 8 cities

position →

N (M - 1)

city ↓ N (M - 1)

k \ i	1	2	3	4	5	6	7	8	9	10
1	1									
2			1							
3						1				
4							1			
5		1								
6									1	
7					1					
8										1
9		0		1						0
10		0						1		0

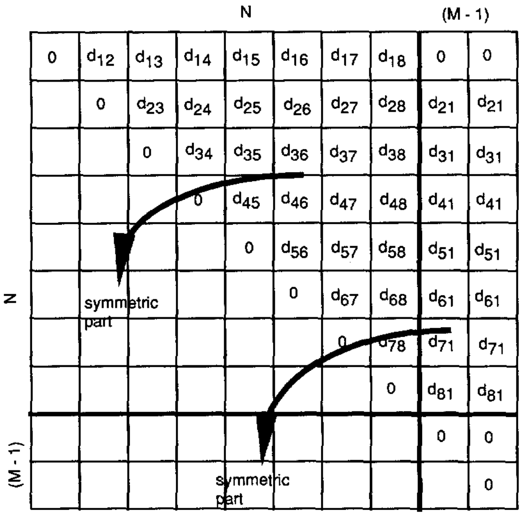
(All of the empty entries in this map are zeros.)

Fig. 2. City-position map

that a fictitious city will not be visited in, at least, r positions before and after the base city.

P3. There should be exactly $(M-1)$ "1"s, which are at least r positions apart from each other, in the last $(M-1)$ rows. This constraint ensures that no subsequent visits of two fictitious cities (and consequently more than two) can be formed and that each salesman, in the original problem, will visit at least r nonfictitious cities.

As an example, consider eight cities and three salesmen (Figs. 1 and 2). Fictitious cities are numbered 9 and 10. One possible permutation matrix is given in Fig. 2. It corresponds to the valid tour shown in Fig. 1.



N (M - 1)

N (M - 1)

0	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	0	0
	0	d_{23}	d_{24}	d_{25}	d_{26}	d_{27}	d_{28}	d_{21}	d_{21}
		0	d_{34}	d_{35}	d_{36}	d_{37}	d_{38}	d_{31}	d_{31}
			0	d_{45}	d_{46}	d_{47}	d_{48}	d_{41}	d_{41}
				0	d_{56}	d_{57}	d_{58}	d_{51}	d_{51}
	symmetric part				0	d_{67}	d_{68}	d_{61}	d_{61}
						0	d_{78}	d_{71}	d_{71}
							0	d_{81}	d_{81}
								0	0
			symmetric part						0

Fig. 3. Modified distance-matrix

The matrix of distances between cities must be modified according to the expansion depicted in Fig. 2. We note that the distance between any city and one of the fictitious cities is equal to the distance between that city and the base city. An example of a modified distance matrix for a transformed eight-city, three-salesmen MTSP is shown in Fig. 3.

3 The Neural Networks Model

In this section, we describe the energy function whose minimization corresponds to solving the MTSP. This function consists of two parts: the physical objective function (i.e., the total length of the salesmen's tours), and the constraints. All constraints are chosen so as to ensure that states corresponding to the desired permutation matrix are favored.

Since we are going to use a $(M+N-1)\times(M+N-1)$ city-position map, a total of $(M+N-1)$ [2] neurons are needed to form a network for the MTSP. Each state of a neuron will be represented by $U_{ki}\in R$, where k denotes a city and i its position in a tour. The output of a neuron is obtained from its state through a sigmoid nonlinearity (the activation function):

$$V_{ki}=\tfrac{1}{2}[1+\tanh(U_{ki}/U_{00})], \qquad (1)$$

where U_{00} is a constant. The value of $V_{ki}\in(0,1)$ expresses the tendency of city k to be located in position i in a tour. A final solution to the problem is obtained whenever the state variables converge to a stable steady state in time

$$\lim_{t\to\infty} V_{ki}(t)=\begin{cases}1 & \text{if city } k \text{ is in position } i\\ 0 & \text{if city } k \text{ is } \textit{not} \text{ in position } i,\end{cases}$$

$$\forall k,i=1,2,\ldots,(N+M-1).$$

The energy function which describes the total length of a tour is given as follows:

$$E_p=\tfrac{1}{2}\sum_{k=1}^{N+M-1}\sum_{\substack{l=1\\l\neq k}}^{N+M-1}\sum_{i=1}^{N+M-1} d_{kl}V_{ki}(V_{l,i+1}+V_{l,i-1}). \quad (2)$$

Here d_{kl} is the distance between city k and l.
We have to solve

$$\underset{\underline{V}(\underline{U})}{\text{minimize}}\ E_p \quad (3)$$

subject to

$$E_1=\tfrac{1}{2}\sum_{k=1}^{N+M-1}\sum_{i=1}^{N+M-1}\sum_{\substack{j=1\\j\neq i}}^{N+M-1} V_{ki}V_{kj}=0, \quad (4)$$

$$E_2=\tfrac{1}{2}\sum_{i=1}^{N+M-1}\sum_{k=1}^{N+M-1}\sum_{\substack{l=1\\l\neq k}}^{N+M-1} V_{ki}V_{li}=0, \quad (5)$$

$$E_3=\tfrac{1}{2}\left[\left(\sum_{k=1}^{N+M-1}\sum_{i=1}^{N+M-1} V_{ki}\right)-(N+M-1)\right]^2=0, \quad (6)$$

$$E_4=\tfrac{1}{2}\sum_{k=N+1}^{N+M-1}\sum_{\substack{l=N+1\\l\neq k}}^{N+M-1}\sum_{i=1}^{N+M-1} V_{ki} \times\sum_{s=1}^{r}(V_{l,i+s}+V_{l,i-s})=0, \quad (7)$$

$$E_5=\tfrac{1}{2}\left[\left(\sum_{k=N+1}^{N+M-1}\sum_{i=1}^{N+M-1} V_{ki}\right)-(M-1)\right]^2=0, \quad (8)$$

$$\begin{aligned}&V_{1,1}=1,\\&V_{1,2}=V_{1,3}=\ldots=V_{1,N+M-1}=0,\\&V_{2,1}=V_{3,1}=\ldots=V_{N+M-1,1}=0,\\&V_{k,2}=V_{k,3}=V_{k,4}=\ldots=V_{k,r+1}=0,\end{aligned} \quad (9a)$$

$$\begin{aligned}V_{k,N+M-r}&=V_{k,N+M+1-r}\\&=V_{k,N+M+2-r}=\ldots=V_{k,N+M-1}=0,\\&\forall k=N+1,\ldots,N+M-1.\end{aligned} \quad (9b)$$

All subscripts are defined modulo $(N+M-1)$. Constraints (4), (5), and (6) are required to form a permutation matrix. They ensure that each city (including the fictitious cities) is visited by only one salesman once. Constraint (6) is equivalent to (4) and (5), yet vital to this algorithm. It is required since (4) and (5) cannot prevent a situation where the whole city-position map is full of zeros. Thus it ensures that any feasible solution to the problem contains $(N+M-1)$ entries of value 1 in the entire city-position map. Constraints (7) and (9) are needed to have city number 1 as a base city and to have exactly M nontrivial closed loops where each one consists of, at least, r customer cities (see properties P2 and P3). Constraint (8) follows from the fact that any feasible solution to the transformed problem contains $(M-1)$ "1"s in the last $(M-1)$ rows of the city-position map, which represent salesmen leaving and returning to the base city 1. This constraint, although redundant in our formulation, has been found to be very useful in the computation. It accelerates the decision making procedure resulting from the network's dynamics.

It is well known that a constrained optimization problem can be converted to an unconstrained one by introducing Lagrangian multipliers, λ_α, and minimizing the resulting energy function [25], i.e.,

$$\underset{\underline{V}(\underline{U})}{\text{minimize}}\quad E=E_p+\sum_{\alpha=1}^{5}\lambda_\alpha E_\alpha. \quad (10)$$

Considering the λ_α as the states of additional five neurons, we can obtain a system of ODEs

$$\frac{\mathrm{d}U_{ki}}{\mathrm{d}t}=-\left[\frac{\partial E_p}{\partial V_{ki}}+\sum_{\alpha=1}^{5}\lambda_\alpha\frac{\partial E_\alpha}{\partial V_{ki}}\right],\quad \forall k,i=1,2,\ldots,N+M-1, \quad (11a)$$

$$\frac{\mathrm{d}\lambda_\alpha}{\mathrm{d}t}=-E_\alpha,\quad \forall\alpha=1,2,\ldots,5 \quad (11b)$$

as a gradient system associated with the energy given in (10). However, it does not guarantee that (11) has a stable equilibrium point that fulfills the problem's constraints (i.e., a feasible solution). To ensure convergence to a stable equilibrium point on the state space boundaries (i.e., $V_{ki}\in\{0,1\}$, $\forall k,i=1,\ldots,N+M-1$), we use *i)* the BDMM suggested by Platt and Barr [26] and *ii)* the damping term, $-U_{ki}/\tau$ (τ constant), suggested by Hopfield and Tank [7] to obtain

$$\begin{aligned}\frac{\mathrm{d}U_{ki}}{\mathrm{d}t}&=-\frac{U_{ki}}{\tau}-\left[\frac{\partial E_p}{\partial V_{ki}}+\sum_{\alpha=1}^{5}\lambda_\alpha\frac{\partial E_\alpha}{\partial V_{ki}}\right]\\&=-\left[\frac{U_{ki}}{\tau}+\frac{\partial E_p}{\partial V_{ki}}\right]-\sum_{\alpha=1}^{5}\lambda_\alpha\frac{\partial E_\alpha}{\partial V_{ki}}\\&\quad\forall k,i=1,2,\ldots,N+M-1,\end{aligned} \quad (12a)$$

$$\frac{\mathrm{d}\lambda_\alpha}{\mathrm{d}t}=+E_\alpha,\quad \forall\alpha=1,2,\ldots,5 \quad (12b)$$

with the condition (9).

Introducing the damping term $(-U_{ki}/\tau)$ is equivalent to adding an extra energy term to (10) of the form

$$E_{HT}=\frac{1}{\tau}\sum_{k=1}^{N+M-1}\sum_{i=1}^{N+M-1}\int_{0.5}^{V_{ki}}U_{ki}(V_{ki})\mathrm{d}V_{ki} \quad (13)$$

such that the problem is to minimize the energy

$$E_{\text{tot}}=E+E_{HT}. \quad (14)$$

The maximum value of this integral, when V_{ki} equals either to 1 or 0, is $0.5U_{00}\log 2$. In the high-gain limit of the activation function (1), $U_{00} \to 0$ and/or for a large value of τ, this term becomes negligible, and the location of the constrained minima of $E+E_{HT}$ can be shown to be the same as the minima of E.

As Hopfield [27] shows, for small, nonzero U_{00}, this term contributes mainly near the state space surfaces and has therefore a tendency to displace the minima slightly toward the interior of the space. However, in a constrained optimization problem, the energy terms due to the constraints compensate for this tendency and force the state variables of the network to move back to the surfaces.

As recommended by Platt and Barr [26], the plus sign is chosen in (12b). Although this results in gradient ascent on the λ_α, it stabilizes the BDMM and forces the solution to fulfill the constraints.

Differentiation of E_p and E_1–E_5 with respect to V_{ki} and substitution into (12a) yield our dynamic model in its final explicit form

$$\begin{aligned}\frac{\mathrm{d}U_{ki}}{\mathrm{d}t} = -\frac{U_{ki}}{\tau}\Bigg\{ & \sum_{\substack{l=1\\ l\neq k}}^{N+M-1} d_{kl}(V_{l,i+1}+V_{l,i-1}) \\ & +\lambda_1 \sum_{\substack{j=1\\ j\neq 1}}^{N+M-1} V_{kj} + \lambda_2 \sum_{\substack{l=1\\ l\neq k}}^{N+M-1} V_{li} \\ & +\lambda_3\left[\left(\sum_{l=1}^{N+M-1}\sum_{j=1}^{N+M-1} V_{lj}\right) - (N+M-1)\right] \\ & +\Bigg(\lambda_4 \sum_{\substack{l=N+1\\ l\neq k}}^{N+M-1} \sum_{s=1}^{r} (V_{l,i+s}+V_{l,i-s}) \\ & +\lambda_5\left[\left(\sum_{l=N+1}^{N+M-1}\sum_{j=1}^{N+M-1} V_{lj}\right) - (M-1)\right]\Bigg)\Bigg\}, \\ & \forall k, i = 1, 2, \ldots, N+M-1, \end{aligned} \tag{15a}$$

$$\frac{\mathrm{d}\lambda_\alpha}{\mathrm{d}t} = +E_\alpha, \quad \forall \alpha = 1, 2, \ldots, 5. \tag{15b}$$

The last two terms in the RHS of (15a) denoted by the index k are taken into account only in the equations of the state variables which belong to the subset $k = \{N+1, \ldots, N+M-1\}$ of the last $(M-1)$ rows of the city-position map.

The connection strengths between the neurons in this network can be retrieved from (9) and (15). Note that the connections between the λ_α neurons and all of the U_{ki} neurons are anti-symmetric.

Cohen and Grossberg [28] have shown that a system of equations of the form (15a) admits a global Lyapunov function under certain conditions. However, these conditions do not apply to our BDMM dynamic model since there are five more coupled ODEs for the λ_α. We are in the process of investigating the stability properties of (15) based on the Lyapunov function candidate suggested by Platt and Barr [26]. Numerical experiments indicate that our model might be stable and converge to valid solutions for a wide range of problem parameters.

4 Simulation Results and Discussion

We have found that the dynamic system of equations that govern the information processing in the network behaves more efficiently as the number of the interacting neurons and their interconnections increases. Therefore, double and triple summation terms of the neurons' output appear in (3)–(8) to express the required connectivity. In addition, the variables are evaluated twice by the summation terms of the energy functions, and then multiplied by 0.5 to obtain the correct "physical" quantity. Computations were carried out on the host computer of the ORNL/CESAR NCUBE hypercube multiprocessor (NCUBE Corporation, Beaverton, Oregon) which operated here as a common sequential computer. No attempt has been made so far to carry out concurrent computation.

The solution of the dynamic model [(15), (1), and (9)] was carried out numerically using the first order explicit Euler numerical method

$$\begin{aligned} U_{ki}^{\mathrm{new}} &= U_{ki}^{\mathrm{old}} + \Delta t\left[\frac{\mathrm{d}U_{ki}}{\mathrm{d}t}\right]^{\mathrm{old}} \\ &= U_{ki}^{\mathrm{old}} - \Delta t[U_{ki}/\tau + R_{ki}]^{\mathrm{old}}, \end{aligned} \tag{16a}$$

$$\lambda_\alpha^{\mathrm{new}} = \lambda_\alpha^{\mathrm{old}} + \Delta t E_\alpha^{\mathrm{old}}. \tag{16b}$$

Here "new" and "old" designate current and previous time step values, respectively, and R_{ki} is the expression in brackets in the RHS of (15a) evaluated at the previous time step. The calculation was carried out synchronously. The neurons' time constant, τ, was chosen without any loss of generality, to be $\tau = 1$. The time stepsize, Δt, in our simulations was between 10^{-5} and 10^{-4} s. As the number of cities increases, smaller values of Δt are needed to retain the numerical stability of this scheme.

The initial conditions which are needed to solve the dynamic model, i.e., $U_{ki}(0)$ and $\lambda_\alpha(0)$ can be chosen arbitrarily. As suggested by Hopfield and Tank [7], the initial conditions of the "neural" state variables, $U_{ki}(0)$, should be chosen without bias in favor of any particular tour. This allows the system to evolve monotonically towards the desired solution. The initial values of the Lagrange multipliers, $\lambda_\alpha(0)$, can be chosen at random but should be positive. Their values affect the number of iterations and therefore the speed with which the solution satisfies the constraints. Selection of negative values for $\lambda_\alpha(0)$, would cause the equations to take longer to converge to a valid tour. We have not yet tried to optimize this choice of initial values. The gain

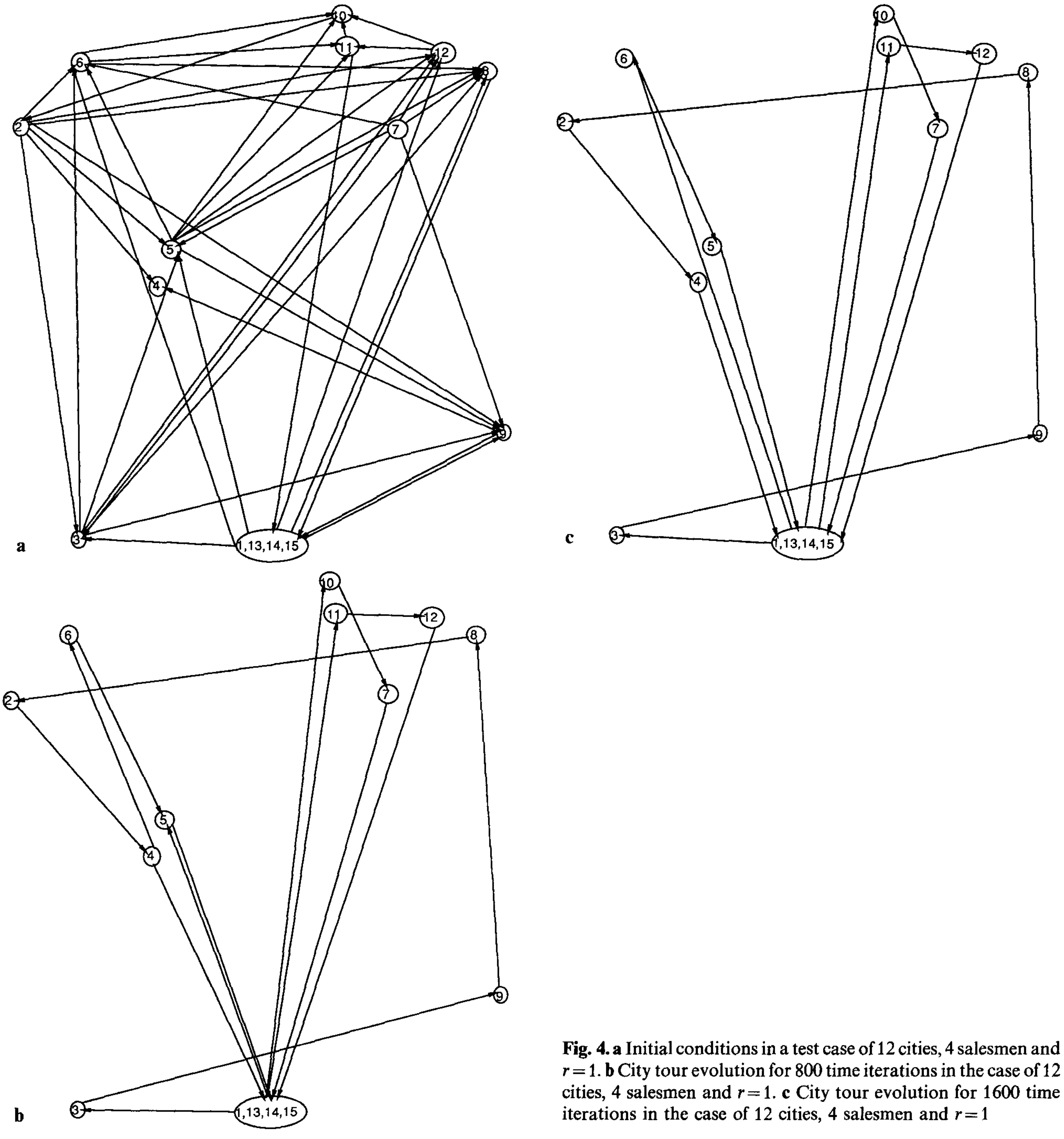


Fig. 4. a Initial conditions in a test case of 12 cities, 4 salesmen and $r=1$. **b** City tour evolution for 800 time iterations in the case of 12 cities, 4 salesmen and $r=1$. **c** City tour evolution for 1600 time iterations in the case of 12 cities, 4 salesmen and $r=1$

U_{00} of the activation function (1) was chosen in the interval [0.02, 0.05].

Two series of tests with N up to 30 were carried out to experimentally verify the validity of the proposed algorithm and its capabilities. First, several runs were made for which all the parameters were kept constant except the number of salesmen which was changed from $M=1$ to $M=N-1$. Second, two series of simulations were performed for $N=12$, $M=4$, and $N=20$, $M=4$ with $r=1$, for which all parameters were kept constant except the initial conditions of the state variables. In all cases tested, convergence to valid tours was observed and the simulated total run time (i.e., Δt^* [number of iterations]) did not exceed the neurons' time constant, $\tau=1$. The locations of the cities were chosen randomly within a unit square.

An example of paths obtained in a case of 12 cities and 4 salesmen, in the first series of experiments, after

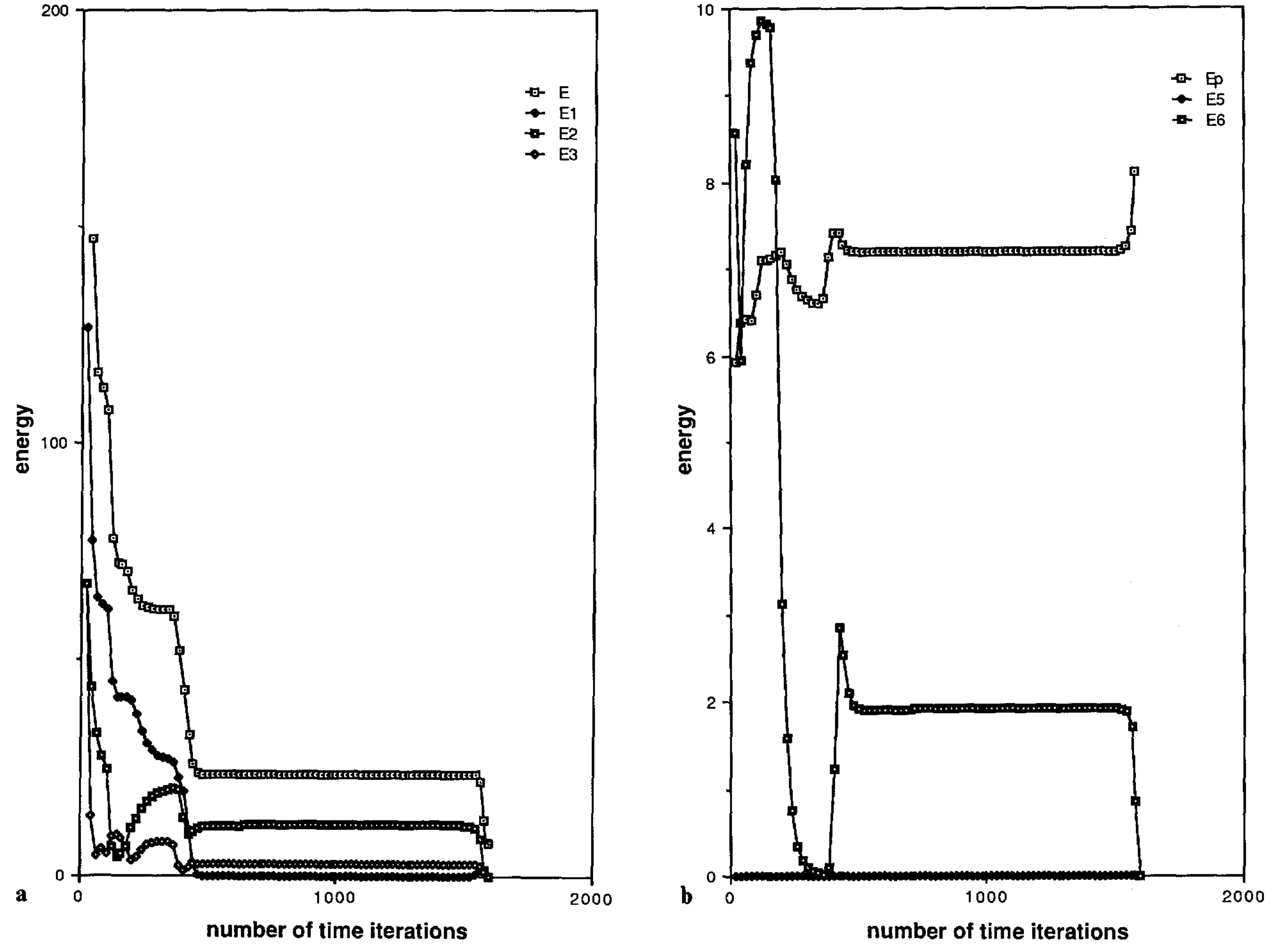


Fig. 5. a Energy variation vs. number of time iterations for the case of 12 cities, 4 salesmen and $r=1$. **b** Energy variation vs. number of time iterations for the case of 12 cities, 4 salesmen and $r=1$

different numbers of iterations is shown in Figs. 4a–c. The initial state described in Fig. 4a represents a state space with a large tendency of multiple visits to each city. Figure 4b describes the state space after 800 iterations. It can be seen that the tours to cities 4, 5, and 6 have not yet been established. The final solution obtained after 800 additional iterations is shown in Fig. 4c.

The variation of the energies E_p, E_1–E_5, and E [(2), (4)–(8), and (10), respectively] versus the number of time iterations for the 12 cities and 4 salesmen experiment is shown in Figs. 5a and 5b. It can be seen how the contributions to the total energy are transferred between the various components while the total energy, E, gradually decreases to its final local minimum. The study of the network dynamics, and the way by which signals propagate in it as a response to different states encountered in the solution, is very much facilitated by examining the evolution of all energy components with time. This led us to realize the necessity to introduce constraint (8) (i.e., E_5).

In the second series of tests, a sample of 50 runs for each case was examined. In both cases (i.e., $N=14$, $M=4$ and $N=20$, $M=4$) Δt was set at 10^{-4} s. All parameters were kept constant except the initial conditions. These were chosen randomly as $U_{ki}(0) = (2 \times rnd - 1)$ where "*rnd*" is a uniform random number between 0 and 1.

In all cases tested, convergence to valid solutions could be observed. Figure 6a and 6b depicts histograms that show the relative number of occurrences of tour lengths and number of iterations required for convergence as a function of the initial conditions. These histograms were generated for the case of 20 cities and 4 salesmen with $r=1$, from 50 random initial conditions samples. Due to small sample size, the results give only crude estimation of the probabilities of occurrence. Examination of total tour lengths in many other instances indicates that the algorithm developed provides good solutions to the problem.

Figure 7 describes the solution obtained for the case of 20 cities and 4 salesmen where each salesman is

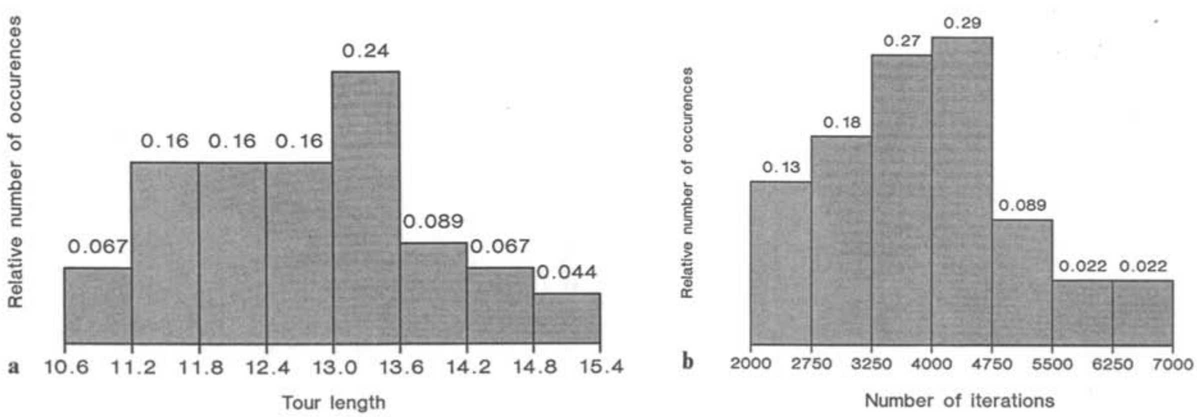


Fig. 6. **a** A histogram of the number of paths of different total distances for the MTSP with 20 cities, 4 salesmen and $r = 1$. **b** A histogram of the number of iterations for the MTSP with 20 cities, 4 salesmen and $r = 1$

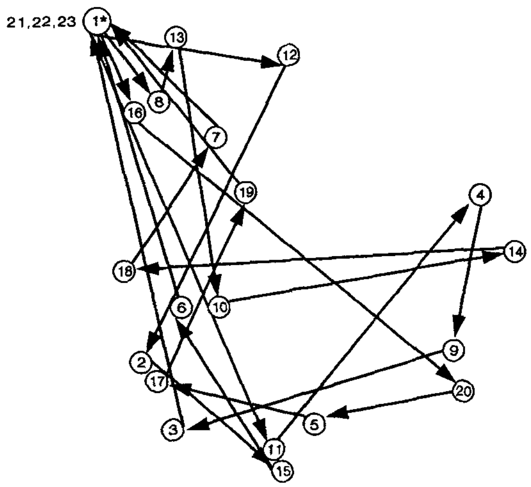


Fig. 7. A feasible solution for an instance of the MTSP with 20 cities, 4 salesmen and $r = 4$

required to visit at least four customer cities (i.e., $r = 4$), subject to a particular set of initial conditions. This solution suggests that each one of the salesmen will carry out a tour which consists of one of the following sequences:

$$\{1, 12, 2, 15, 6, 1(21)\}, \quad \{1(21), 8, 13, 10, 14, 18, 7, 1(22)\},$$

$$\{1(22), 16, 20, 5, 17, 19, 1(23)\}, \text{ and } \{1(23), 11, 4, 9, 3, 1\}.$$

Numbers 21, 22, and 23 inserted in parentheses indicate the corresponding fictitious city numbers in this solution.

5 Summary and Conclusions

A neural network algorithm for the solution of the MTSP was developed. The solution is based on a transformation of the MTSP into the standard TSP. The transformed problem is represented in the neural state variables domain by an expanded version of the city-position map suggested by Hopfield and Tank [7]. The expanded map is referred to $(N + M - 1)$ cities and a single fictitious salesman. The dynamic model associated with the problem is an extension of the Hopfield and Tank approach incorporating the Basic Differential Multipliers Method [26]. In this model, five additional ODEs for the Lagrange multipliers, λ_α, are solved simultaneously with $(N + M - 1)$ state equations. The first order explicit Euler method was used for the numerical solution of the dynamic model.

Two series of tests, with N up to 30, were carried out to experimentally verify the validity of the proposed algorithm and its capabilities. In all cases tested, convergence to valid tours was achieved. Time stepsizes between 10^{-5} and 10^{-4} s were used in the numerical simulations. The number of iterations required to find a solution varied between 240 and 5500. Simulations of two cases ($N = 12$, $M = 4$, and $N = 20$, $M = 4$) have indicated that the model can provide good solutions in a reasonable amount of iterations.

Run time of the algorithm on standard computers was not an important issue in our research. The incentive for the development of this model was its adequacy for implementation in fast analog circuits. The results showed that the time required for convergence in all cases tested was less than one characteristic time of the neurons ($\tau = 1$) that make up the network. As mentioned by Hopfield and Tank, one may expect convergence times of 10–100 ms for networks of biological neurons, while semiconductor circuits could converge in 10^{-5} to 10^{-7} s.

The success of this algorithm in converging to valid tours seems to be due to: *(i)* the use of BDMM, and *(ii)* the introduction of constraint (8). Studies of the network dynamic response during simulations showed that both *(i)* and *(ii)* forced the state variables to change according to the energy gradient, consistently towards the state variables surface. General stability analysis of this model is currently under investigation. This algorithm can be applied to achieve real-time decision making.

Acknowledgements. The authors thank Mr. J. B. Fuqua, whose bright insights improved the simulations significantly, and Stephanie Raby for carefully typing the manuscript. Special thanks go to Dr. C. R. Weisbin and Dr. J. Barhen for their continuous encouragement and support. One of the authors (J. Han) was supported in part by an ORNL Postgraduate Research Appointment administered by Oak Ridge Associated Universities. Research was sponsored by the Office of Military Applications and the Office of Basic Energy Sciences, U.S. Department of Energy, under Contract No. DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc., and by the U.S. Air Force Wright Aeronautical Laboratories, under Interagency Agreement DOE-40-1579-85

References

1. Angel RD, Candle WL, Noonan R, Whinston A (1972) Computer assisted school bus scheduling. Manag Sci 18:279–288
2. Orloff DS (1974) Routing a fleet of *M* vehicles to/from a central facility. Networks 4:147–162
3. Christofides N, Eilon S (1969) An algorithm for vehicle dispatching problems. Oper Res Quart 20:309–318
4. Gavish B (1982) Topological design of centralized computer networks – formulations and algorithms. Networks 12:355–377
5. Gavish B (1983) Formulations and algorithms for capacitated minimal directed tree problems. J Assoc Comput Mach 30:118–132
6. Gavish B, Srikanth KN (1979) Mathematical formulations for the dial-a-ride problem. Graduate School of Management, University of Rochester (working paper, March 1979)
7. Hopfield JJ, Tank DW (1985) "Neural" computation of decision in optimization problems. Biol Cybern 52:141–152
8. Davis GW, Ansari A (1986) Sensitivity analysis of Hopfield neural net. IEEE First Conf Neural Networks 3:325–329
9. Bagherzadeh N, Kerola T, Leddy B, Brice R (1986) Parallel execution of the Traveling Salesman Problem. First Int Conf Neural Networks 3:317–325
10. Parsi Behzad, Behroog Kamangar-Parsi (1986) An efficient neural network for optimization. IEEE First Int Conf Neural Networks 3:785–790
11. Cervants JH, Hildebrant RR (1986) Comparison of three neuron-based computation schemes. IEEE First Int Conf Neural Networks 3:657–671
12. Simmes SD (1986) Program parallel computers using energy minimization algorithms. IEEE First Int Conf Neural Networks, 3
13. Wilson GW, Pawley GS (1988) On the stability of the Traveling Salesman Problem algorithm of Hopfield and Tank. Biol Cybern 58:63–70
14. Toomarian N (1988) A concurrent neural network algorithm for the Traveling Salesman Problem. Third Conf Hypercube Concurrent Comput Appl Pasadena, CA (January 19–20, 1988) (submitted for publication)
15. Posner E (1987) Workshop on Neural Network Devices and Applications, Jet Propulsion Laboratory, Pasadena, CA (February 18–19, 1987)
16. Denker JS (ed) (1986) Neural networks for computing. Am Inst Phys Conf Proc NY 151
17. Mead CA (1988) Analog VLSI and neural systems. Addison-Wesley, Reading, Mass: (to be published)
18. Kirkpatrick S, Toulause G (1988) Configuration space analysis of Traveling Salesman Problems. J Phys 46:1277–1292
19. Skiscim CC, Golden BL (1983) Optimization by simulated annealing: a preliminary computational study for the TSF. S. Roberts, J. Banks, B. Schneiser (ed) Proc 1983 Winter Simul Conf
20. Cerny V, Dreyfus SG (1988) Thermodynamical approach to the Traveling Salesman Problem: an efficient simulation algorithm. J Opt Theor Appl 1(45) 1988
21. Bonomi E, Lutton J (1984) The *N*-city Traveling Salesman Problem: statistical mechanics and the metropolis algorithm. SIAM Rev 4(26) (October 1984)
22. Spurlas N (1986) Statistical mechanics and the Traveling Salesman Problem. Europhys Lett 12:919–923
23. Felten E, Karlin S, Otto SW (1985) The Traveling Salesmen Problem on a hypercube MIMD computer. Proceedings of the 1985 Conference on Parallel Processing
24. Szu H (1986) Fast simulated annealing. In: Denker J (ed) Neural networks for computing. American Institute of Physics, New York, pp 420–425
25. Hestenes MR (1975) Optimization theory. Wiley, New York
26. Platt JC, Barr AH (1988) Constrained differential optimization. Proceedings of the IEEE 1987 NIPS Conference: (to be published)
27. Hopfield JJ (1984) Neurons with graded response have collective computational properties like those of two-state neurons. Proc Natl Acad Sci 31:3088–3092
28. Cohen MA, Grossberg S (1983) Absolute stability of global pattern formulation and parallel memory storage by competitive neural networks. IEEE Trans SMC-13:815–826
29. Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco

Received: April 12, 1988
Accepted in revised form: December 12, 1988

Dr. R. C. Mann
Oak Ridge National Laboratory
Building 6025, MS-6364
P.O. Box 2008
Oak Ridge, TN 37831-6364
USA