Simple "Neural" Optimization Networks: An A/D Converter, Signal Decision Circuit, and a Linear Programming Circuit

DAVID W. TANK AND JOHN J. HOPFIELD

Abstract —**We describe how several optimization problems can be rapidly solved by highly interconnected networks of simple analog processors. Analog-to-digital (A/D) conversion was considered as a simple optimization problem, and an A/D converter of novel architecture was designed. A/D conversion is a simple example of a more general class of signal-decision problems which we show could also be solved by appropriately constructed networks. Circuits to solve these problems were designed using general principles which result from an understanding of the basic collective computational properties of a specific class of analog-processor networks. We also show that a network which solves linear programming problems can be understood from the same concepts.**

I. Introduction

We HAVE shown in earlier work [1], [2] how highly interconnected networks of simple analog processors can collectively compute good solutions to difficult optimization problems. For example, a network was designed to provide solutions to the traveling salesman problem. This problem is of the *np*-complete class [3] and the network could provide good solutions during an elapsed time of only a few characteristic time constants of the circuit. This computation can be considered as a rapid and efficient contraction of the possible solution space. However, a globally optimal solution to the problem is not guaranteed; the networks compute locally optimal solutions. For the traveling salesman problem, even among the extremely good solutions, the topology of the optimization surface in the solution space is very rough; many good solutions are at least locally similar to the best solution, and a complicated set of local minima exist. In difficult problems of recognition and perception, where rapidly calculated good solutions may be more beneficial than slowly computed globally optimal solutions, collective computation in circuits of this design may be of practical use.

We have recently found that several less complicated optimization problems which are not of the *np*-complete class can be solved by networks of analog processors. The two circuits described in detail here are an A/D converter and a circuit for solving linear programming problems. These networks are guaranteed of obtaining globally optimal solutions since the solution spaces (in the vicinity of specific initial conditions) have no local minima. The A/D converter is actually one simple example of a *class* of problems for which appropriately constructed collective networks should rapidly provide good solutions. The general class consists of signal decomposition problems in which the goal is the calculation of the optimum fit of an integer coefficient combination of basis functions (possibly a nonorthogonal set) to an analog signal. The systematic approach we have developed to design such networks should be more broadly applicable.

Fahlman [4] has suggested a rough classification of parallel-processor architectures based upon the complexity of the messages that are passed between processing units. At the highest complexity are networks in which each processor has the power of a complete von Neumann computer, and the messages which are passed between individual processors can be complicated strings of information. The simplest parallel architectures are of the "value-passing" type. Processor-to-processor communication between local computations consists of a single binary or analog value. The collective analog networks considered here are in this class; each processor makes a simple computation or decision based upon its analysis of many analog values (information) it receives in parallel from other processors in the network. Our motivation for studying the computational properties of circuits with this organization arose from an attempt to understand how known biophysical properties and architectural organization of neural systems can provide the immense computational power characteristic of the brains of higher animals. In our theoretical modeling of neural circuits [1], [2], [5], [6], each neuron is a simple analog processor, while the rich connectivity provided in real neural circuits by the synapses formed between neurons are provided by the parallel communication lines in the value-passing analog processor networks. Hence, in addition to designs for conventional implementation with electrical components, the circuits and design principles described here add to the known repertoire of neural circuits which seem neurobiologically plausible. In general, a consideration of such circuits provides a methodology for assigning function to anatomical structure in real neural circuits.

Manuscript received August 27, 1985; revised This work was supported in part by the National Science Foundation under Grant PCM-8406049.

D. W. Tank is with the Molecular Biophysics Research Department, AT&T Bell Laboratories, Murray Hill, NJ 07974.

J. J. Hopfield is with the Division of Chemistry and Biology, California Institute of Technology, Pasadena, CA 91125.

IEEE Log Number 8607497.

0098-4094/86/0500-0533$01.00 ©1986 IEEE

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

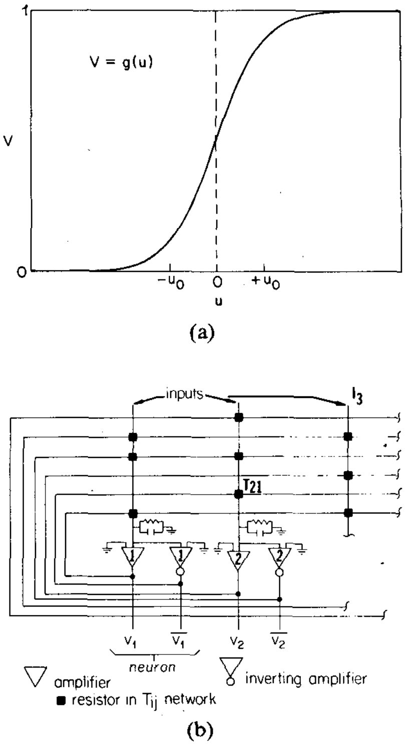
II. The A/D Converter Network

We have presented in detail [1], [2], [5] the basic ideas involved in designing networks of analog processors to solve specific optimization problems. The general structure of the networks we have studied is shown in Fig. 1(b). The processing elements are modeled as amplifiers having a sigmoid monotonic input–output relation, as shown in Fig. 1(a). The function $V_j = g_j(u_j)$ which characterizes this input–output relation describes the output voltage V_j of amplifier j due to an input voltage u_j. The time constants of the amplifiers are assumed negligible. However, each amplifier has an input resistor leading to a reference ground and an input capacitor. These components partially define (see [1] and [5]) the time constants of the network and provide for integrative analog summation of input currents from other processors in the network. These input currents are provided through resistors of conductance T_{ij} connected between the output and amplifier j and the input of amplifier i. In order to provide for output currents of both signs from the same processor, each amplifier is given two outputs, a normal output, and an inverted output. The minimum and maximum outputs of the normal amplifier are taken as 0 and 1, while the inverted output has corresponding values of 0 and -1. A connection between two processors is defined by a conductance T_{ij} which connects one of the two outputs of amplifier j to the input of amplifier i. This connection is made with a resistor of value $R_{ij} = 1/|T_{ij}|$. (In Fig. 1, resistors connecting 2 wires are schematically indicated by squares.) If $T_{ij} > 0$, this resistor is connected to the normal output of amplifier j. If $T_{ij} < 0$, it is connected to the inverted output of amplifier j. The matrix T_{ij} defines the connectivity among the processors. The net input current to any processor (and hence the input voltage u_i) is the sum of the currents flowing through the set of resistors connecting its input to the outputs of the other processors. Also, as indicated in Fig. 1(b), externally supplied input currents (I_i) are also present for each processor. In the circuits discussed here, these external inputs can be constant biases which effectively shift the input–output relation along the u_i axis and/or problem-specific input currents which correspond to data in the problem.

We have shown [5] that in the case of symmetric connections ($T_{ij} = T_{ji}$), the equations of motion for this network of analog processors always lead to a convergence to *stable states*, in which the output voltages of all amplifiers remain constant. Also, when the diagonal elements (T_{ii}) are 0 and the width of the amplifier gain curve (Fig. 1(a)) is narrow —the high-gain limit—the stable states of a network comprised of N processors are the local minima of the quantity

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} V_iI_i. \qquad (1)$$

We refer to E as the computational energy of the system. By construction, the state space over which the circuit operates is the *interior* of the N-dimensional hypercube defined by $V_i = 0$ or 1. However, we have shown that in the high-gain limit networks with vanishing diagonal connections (T_{ii}) = 0 have minima only at *corners* of this space [5]. Under these conditions the stable states of the network correspond to those locations in the discrete space consisting of the 2^N corners of this hypercube which minimize E (1). (Somewhat less restrictive conditions will often suffice, which allow leeway for nonzero, T_{ii}. Negative T_{ii} do not necessarily cause problems.)



(a)

(b)

Fig. 1. (a) The input–output relation for the processors (amplifiers) in Fig. 1(b). (b) The network of analog processors. The output of any neuron can potentially be connected to the input of any other neuron. Black squares at intersections represent resistive connections (T_{ij}'s) between outputs and inputs. Connections between inverted outputs and inputs represent negative (inhibitory) connections.

Networks of analog processors with this basic organization can be used to compute solutions to specific optimization problems by relating the minimization of the problems cost function to the minimization of the E function of the network. Since the energy function can be used to define the values of the connectivities (T_{ij}) and input bias currents (I_i), relating a specific problem to a specific E function provides the information for a detailed circuit diagram for the network which will compute solutions to the problem. The computation consists of providing an initial set of amplifier input voltages u_i, and then allowing the analog system to converge to a stable state which minimizes the E function. The solution to the problem is then interpreted from the final stable state using a predetermined rule.

The A/D converter we shall describe is a specific example of such an optimization network. For clarity, we will limit the present discussion to a 4-bit converter. Its wiring diagram is shown in Fig. 2. The circuit consists of 4 amplifiers (only inverting outputs are needed—see below) whose output voltages will be decoded to obtain the output

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

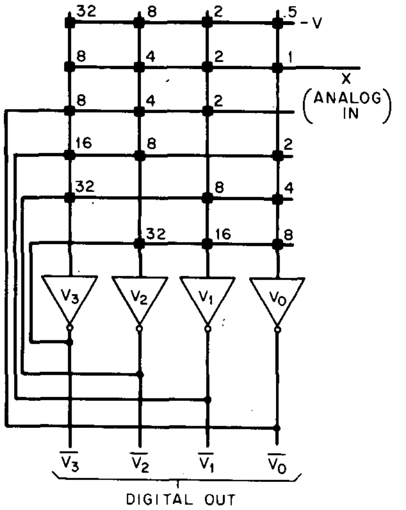


Fig. 2. The 4-bit A/D converter computational network. The analog input voltage is x, while the complement of the digital word $V_3V_2V_1V_0$ which is computed to be the binary value of x is read out as the 0 or 1 values of the amplifier output voltages.

binary word of the converter, a network of feedback resistors connecting the outputs of one amplifier to the inputs of the others, a set of resistors (top row) which feed different constant current values into the input lines of the amplifiers, and another set of resistors (second row) which inject current onto the input lines of the amplifiers which are proportional to the analog input voltage x, which is to be converted by the circuit. For the present we assume that the output voltages (V_i) of the amplifiers can range between a minimum of 0 V and a maximum of 1 V. Thus as described above for the variables in (1), the V_i range over the domain [0,1]. We further assume that the value of x in volts is the numerical value of the input which is to be converted. The converter network is operating properly when the integer value of the binary word represented by the output states of the amplifiers is numerically equal to the analog input voltage. In terms of the variables defined above, this criterion can be written as

$$\sum_{i=0}^{3} V_i 2^i \approx x. \tag{2}$$

The circuit of Fig. 2 is organized so that this expression always holds.

The strategy employed in creating this design is to consider A/D conversion as a simple example of an optimization problem. If the word $V_3V_2V_1V_0$ is to be the "best" digital representation of x, then two criteria must be fulfilled. The first is that each of the V_i have the value of 0 or 1, or at least be close enough to these values so that a separate comparator circuit can establish digital logic levels. The second criterion is that the particular set of 1's and 0's chosen is that which "best" represents the analog signal. This second criterion can be expressed, in a least-squares sense, as the choice of V_i which minimize the energy function

$$E = \frac{1}{2}\left(x - \sum_{i=0}^{3} V_i 2^i\right)^2 \tag{3}$$

because the quadratic is a minimum when the parenthesized term has a minimum absolute value. If this function is expanded and rearranged, it can be put in the form of (1) (plus a constant). There would, therefore, be a real circuit of the class shown in Fig. 1 which would compute by trying to minimize (3).

However, with this simple energy function there is no guarantee that the values of V_i will be near enough to 0 or 1 to be identified as digital logic levels. Since (3) contains diagonal elements of the T-matrix of the form $\alpha(V_i)^2$ which are nonzero, the minimal points to the E function (3) will not necessarily lie on the corners of the space, and thus represent a digital word (see [5]). Since there are many combinations of the V_i which can be linearly combined to obtain x, a minimum can be found which is not at a corner of the space.

We can eliminate this problem by adding one additional term to the E function. Its form can be chosen as

$$-\frac{1}{2}\sum_{i=0}^{3}(2^i)^2[V_i(V_i-1)]. \tag{4}$$

The *structure* of this term was chosen to favor digital representations. Note that this term has minimal value when, for each i, either $V_i = 1$ or $V_i = 0$. Although any set of (negative) coefficients will provide this bias towards a digital representation, the *coefficients* in (4) were chosen so as to cancel out the diagonal elements in (3). The elimination of diagonal connection strengths will generally lead to stable points only at corners of the space. The term (4) equally favors *all* corners of the space, and does not favor any particular digital answer. Thus the total energy E which contains the sum of the two terms in (3) and (4) has minimal value when the V_i are a *digital* representation close to x.

This completes the energy function for the A/D converter. It can be expanded and rearranged into the form

$$E = -\frac{1}{2}\sum_{j=0}^{3}\sum_{i \neq j=0}^{3}(-2^{i+j})V_iV_j - \sum_{i=0}^{3}(-2^{(2i-1)}+2^i x)V_i. \tag{5}$$

This is of the form of (1) if we identify the connection matrix elements and the input currents as

$$T_{ij} = -2^{(i+j)}$$
$$I_i = (-2^{(2i-1)}+2^i x). \tag{6}$$

The complete circuit for this 4-bit A/D converter with components as defined above is the network shown in Fig. 2. The inverting output of each amplifier is connected to the input of the other amplifiers through a resistor of conductance 2^{i+j}. The other input currents to each amplifier are provided through resistors of conductance 2^i connected to the input voltage x and through resistors of conductance $2^{(2i-1)}$ connected to a -1-V reference potential. These numbers for the resistive connections on the feedback network and the input lines represent the ap-

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

propriate *relative* conductances of the components and assume that the constant terms in the input currents are provided by connecting the input lines through resistors to a -1-V reference potential, that the minimum and maximum output voltages of the amplifiers are to be 0 and 1 V, and that the analog input voltage to be digitized is in the range $(-0.5, 15.5)$ V. When building a real circuit, the values of the resistors chosen should satisfy the relative conductances indicated in the figure and in the above equations, but their absolute values will depend upon the real voltage rails of the amplifiers, the specific input voltage range to be digitized, and reasonable values for the power dissipation. If the real output voltage range for the amplifiers is $[0, V_{BB}]$, the voltage range to be digitized is $[0, V_H]$, and the reference voltage to be used for the constant input currents is $-V_R$, then it is straightforward to show that the relative conductances (which must now only be scaled for power dissipation) for the feedback connections are

$$T_{ij} = -\frac{2^{(i+j)}}{V_{BB}}$$

while the input voltage x will be fed into the ith amplifier through a resistor of conductance $2^{(4+i)}/V_H$, and the constant current is provided through resistors of conductance $(2^{(i-1)} + (2^{(2i-1)}/V_R))$ connected to the $-V_R$ reference voltage.

The ability of the network to compute the correct digital representation of x was studied in a series of computer experiments and actual circuit construction. In the computer experiments, the dynamic behavior of the network was simulated by integration of the differential equations which describe the circuit (for details, see [1], [5]). The convergence of the network was studied as a function of the analog input x, for 160 different values contained in the interval -0.5 to 15.5 V. The digital solutions computed at a fixed value of x depend upon the initial conditions of the network. These initial conditions are defined by the input voltages (u_i) on the amplifiers at the time that the calculation is initiated. In Fig. 3 is plotted the value of the binary word $V_3V_2V_1V_0$ computed by the network as a function of the value of $(x+0.5)$ for the initial conditions $u_i = 0$. The response is the staircase function characteristic of an A/D converter. In a real circuit, separate electronics which would ground the input lines of the amplifiers before each convergence would be required to implement the initial conditions $(u_i = 0)$ used in these simulations. If the input lines are not zeroed before each calculation, then the circuit exhibits hysteresis as the input voltage x is being continuously varied. For example, if x is slowly turned up through the same series of values used in the calculation of Fig. 3, but, instead of zeroing the input lines before a simulated convergence, we allow the u_i to retain the values stabilized at the end of the previous calculation, we obtain the response shown in Fig. 4. Slowly turning down the x input from its maximum value would provide a response which is the "inverse" of Fig. 4. (The value for any x, in the experiment with x descending, is equal to 15.0 minus the value for $(16.0 - x)$ in the experiment with x ascending.) Some stable states of the network are skipped under this set of initial conditions.

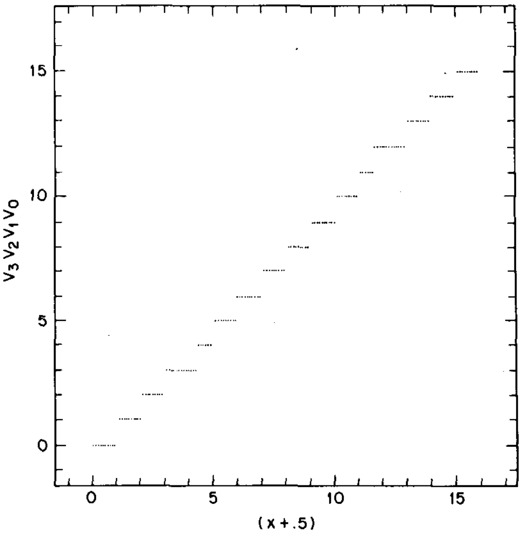


Fig. 3. The digital word computed in simulations of the circuit shown in Fig. 2 as a function of the analog input voltage x. The initial conditions for each of the calculations is $u_i = 0$, for all i.

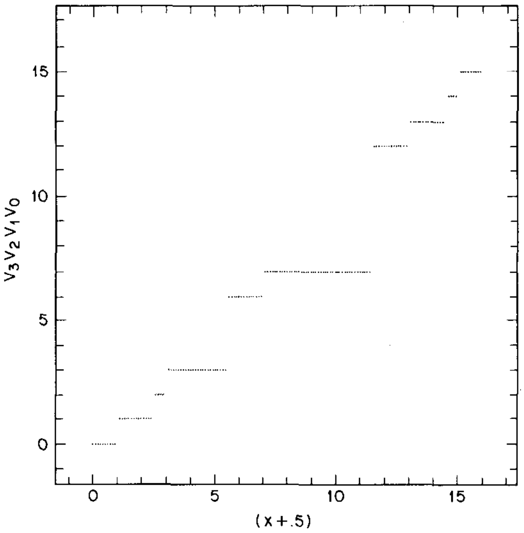
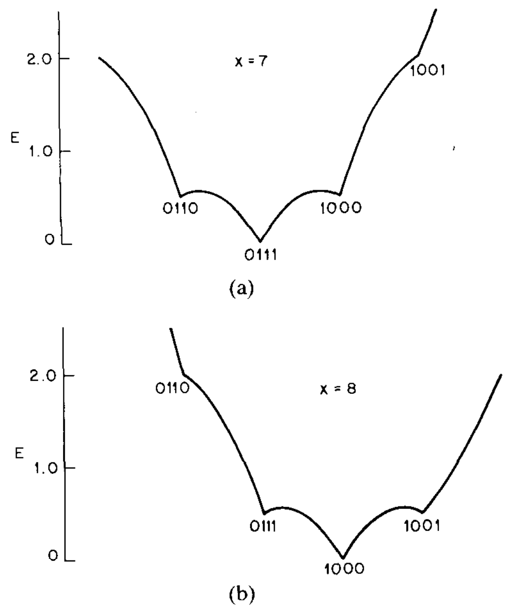


Fig. 4. The results of a calculation similar to that described in Fig. 3 except that the initial conditions were determined by the u_i which stabilized during the previous calculation. Calculations were performed with monotonically increasing values of the analog input voltage x, starting at $x = 0$ V.

One can understand this hysteresis, and its absence for the $u_i = 0$ initial conditions, by considering the topology of the energy surface for fixed x and how it changes as x is varied. In Fig. 5 is shown a *stylized* representation of the energy surface for two different x values. The energy at specific locations in state space is represented, with energy value along the vertical axis. Different corners of state space near the global minimum in E (with value $E = 0$) are indicated along the curve by the set of indices $V_3V_2V_1V_0$. As shown in Fig. 5(a), the energy function for $x = 7$ V has a deep minimum at the corner of state space which is the digital representation of 7, and has local minima at higher E values at the digital representations of 6 and 8. Al-

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.



(a)

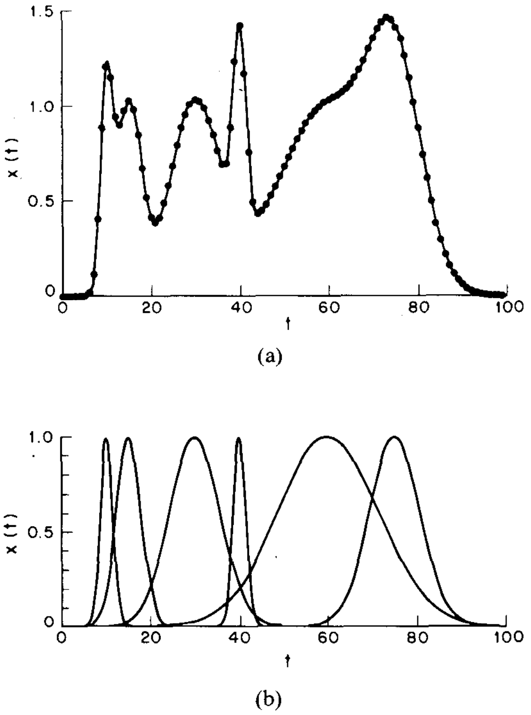
(b)

Fig. 5. A schematic drawing of the energy surface in the vicinity of the global minima for two analog input voltages.

though, as shown above (Fig. 3), the circuit dynamics can lead from a location in state space corresponding to all $u_i = 0$ to that corresponding to the deep minima, if x is changed to 8 V while the u_i are at the $x = 7$ V corner, then although the energy surface will change to that as shown in Fig. 5(b), the system will remain stuck in the now-local minima at the corner corresponding to 7. However, if the circuit is again allowed to compute from the initial conditions $u_i = 0$, but now with $x = 8$ V, the correct deep minima can be obtained. The local minima are a direct consequence of the term (4) in the E function which forces the output voltages to be digital. If this term were not present, the V_i will still represent a valid set of coefficients for the linear approximation of the sum (2) to the analog value x, but the solution will in general not be at one of the corners of the solution space.

III. The Decomposition/Decision Problem

Many problems in signal processing can be described as the attempt to detect the presence or absence of a waveform having a known stereotyped shape and amplitude in the presence of other waveforms and noise. Circuits which are similar to that described above for the A/D converter can be constructed for which the minimal energy state corresponds to a decision about this signal decomposition problem. For example, consider the problem of decomposing a time-dependent analog signal which results from the temporal linear summation of overlapping stereotype Gaussian pulses of known but differing width. A typical summed signal is shown in Fig. 6(a). In Fig. 6(b) is shown the individual pulses which when added together give the signal in Fig. 6(a). The decomposition/decision problem is to determine this particular decomposition of the signal in Fig. 6(a), given the knowledge of the individual stereotype forms. To make the problem specific, we assume that $N \cong 100$ time points of analog data ($x(i)$; $i, \cdots, N$) have been recorded, as indicated by the filled circles in Fig. 6(a), and that the set of basis functions defining the possible "pulses" in Fig. 6(b) are the Gaussian functions of the form



(a)

(b)

Fig. 6. (a) Analog signal comprised of a linear summation of Gaussian pulses of different width and peak location. The pulses summed in (a) are explicitly illustrated in (b).

$$\epsilon_{\sigma t}(i) = e^{-[(i-t)/\sigma]^2}. \tag{10}$$

We will let the width parameter σ take on a finite number of possible values, while the peak position (t) of the pulse can be at any one of the N time points. Since the basis set is specified by the width and peak position parameters, the amplifiers used in the decomposition/decision network can be conveniently indexed by the double-set of indices σ, t. In describing the decomposition, each of these basis functions will have a digital coefficient ($V_{\sigma t}$) which corresponds to the output of an amplifier in the network and which represents the presence or absence of this function in the signal to be decomposed. An energy function which defines an analog computational network and which is minimum when this decomposition/decision problem is solved is

$$E = \frac{1}{2} \sum_{i=1}^{N} \left(x_i - \sum_{\sigma=\sigma_1}^{\sigma_{\max}} \sum_{t=1}^{N} V_{\sigma t} \epsilon_{\sigma t}(i) \right)^2 - \frac{1}{2} \sum_{i=1}^{N} \sum_{\sigma=\sigma_1}^{\sigma_{\max}} \sum_{t=1}^{N} (\epsilon_{\sigma t}(i))^2 [V_{\sigma t}(V_{\sigma t} - 1)] \tag{11}$$

with the basis functions as defined in (10). This expression is of the form (1) and, therefore, defines a set of connection strengths ($T_{\sigma t, \sigma' t'}$) and input currents ($I_{\sigma t}$) for each ampli-

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

fier, with

$$T_{\sigma t,\sigma' t'} = \sum_{i=1}^{N} e^{-[[(i-t)/\sigma]^2+[(i-t')/\sigma']^2]} \tag{12}$$

$$I_{\sigma t} = \sum_{i=1}^{N} x_i e^{-[(i-t)/\sigma]^2} + \frac{1}{2}\sum_{i=1}^{N} e^{-2[(i-t)/\sigma]^2}. \tag{13}$$

A schematic diagram of this computational network is shown in Fig. 7. The signals x_i enter the network in parallel (for a time-varying signal this could be accomplished with a delay line) and produce currents in the input lines of the amplifiers through resistors which define the ith "convolution" component in the expression (13) above. A single resistor for each input connected to a reference voltage can provide the constant bias terms.

The energy function presented above for a Gaussian pulse decomposition/decision circuit can be generalized. If $\vec{\epsilon}_k$; $k=1,\cdots,n$ are a set of basic functions which span the signal space $\vec{X}$, then consider the function

$$E = \frac{1}{2}\left(\vec{X} - \sum_k V_k\vec{\epsilon}_k\right)\cdot\left(\vec{X} - \sum_k V_k\vec{\epsilon}_k\right) - \frac{1}{2}\sum_k(\vec{\epsilon}_k\cdot\vec{\epsilon}_k)\left[V_k(V_k-1)\right]. \tag{7}$$

This function describes a network which has an energy minimum (with $E=0$) when the "best" digital combination of basis functions are selected (with $V_i=1$) to describe the signal. The expression (7) can be expanded and rearranged to give

$$E = \frac{1}{2}\sum_k\sum_{k'\neq k}(\vec{\epsilon}_k\cdot\vec{\epsilon}_{k'})V_kV_{k'} - \sum_k\left[(\vec{X}\cdot\vec{\epsilon}_k) + \frac{1}{2}(\vec{\epsilon}_k\cdot\vec{\epsilon}_k)\right]V_k + \frac{1}{2}(\vec{X}\cdot\vec{X}). \tag{8}$$

This is a function which is comprised of terms which are linear and quadratic in the V_k's. It is, therefore, of the form (1) (plus a constant), if we define

$$T_{kk'} = -(\vec{\epsilon}_k\cdot\vec{\epsilon}_{k'})$$
$$I_k = \left[(\vec{X}\cdot\vec{\epsilon}_k) + \frac{1}{2}(\vec{\epsilon}_k\cdot\vec{\epsilon}_k)\right]. \tag{9}$$

Hence, for the general decomposition/decision problem mapped onto the computational network in Fig. 1, the connection strengths between amplifiers correspond to the dot products of the corresponding pairs of basis functions while the input currents correspond to the convolution of the corresponding basis function with the signal and the addition of a constant bias term.

The A/D converter described earlier can be seen to be a simple example of this more general circuit. In the A/D case, the signal is one-dimensional and consists of only an analog value sampled at a single time point. The basis functions are the values 2^n; $n=0,\cdots,(n-1)$ which are a complete set over the integers in the limited domain $[0,2^n-1]$. The binary word output of the circuit is comprised of the coefficients which describe a linear summation of the basis functions which is closest, in the least squares sense, to the input signal.

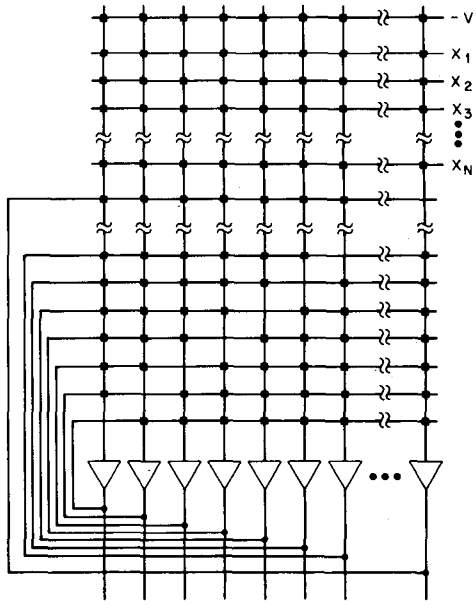


Fig. 7. The general organization of a computational network which can be used to solve a multipoint decomposition problem with nonorthogonal basis functions. The outputs of each of the amplifiers represents the presence ($V_{\sigma,t}=1$) or absence ($V_{\sigma,t}=0$) of a pulse of a width σ and peak location t in the signal trace.

For the A/D converter problem and the Gaussian decomposition/decision network just described, the basis functions which span the signal space are *not* orthogonal. For an orthogonal set, by definition, the connection strengths (9) would all vanish. For example, if the signal consists of N analog-sampled points of a differentiable function, and the basis functions were sines and cosines (a Fourier decomposition network), then the computational circuit would have no feedback connections since these basis functions are orthogonal. In this case, the independent computations made by each amplifier are the convolution of the signal with the particular basis function represented. This is just the familiar rule for calculating Fourier coefficients—all decisions are independent. In general, one can interpret the connections strengths in the decomposition/decision networks as the possible effect of one decision being tested (V_i) on another (V_j); these effects should be zero for orthogonal basis functions.

IV. The Linear Programming Network

The linear programming problem can be stated as the attempt to minimize a cost function

$$\pi = \vec{A}\cdot\vec{V} \tag{14}$$

where $\vec{A}$ is an N-dimensional vector of coefficients for the N variables which are the components of $\vec{V}$. This minimization is to be accomplished subject to a set of M linear

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

constraints among the variables:

$$\vec{D}_j \cdot \vec{V} \geq B_j, \qquad j = 1, \cdots, M$$

$$\vec{D}_j = \begin{bmatrix} D_{j1} \\ D_{j2} \\ \vdots \\ D_{jN} \end{bmatrix} \tag{15}$$

where the $\vec{D}_j$, for each j, contain the N variable coefficients in a constraint equation and the B_j are the bounds. Although we know of no way to directly cast this problem into the explicit form of (1) so that a network of the form shown in Fig. 1 could be used to compute solutions to the problem, we can understand how the circuit in Fig. 8, illustrated for the specific case of two variables ($N = 2$) and four constraint equations ($M = 4$), can rapidly compute the solution to this optimization problem, by a variation of a mathematical analysis used earlier [5].

In the circuit of Fig. 8, the N outputs (V_i) of the left-hand set of amplifiers will represent the values of the variables in the linear programming problem. The components of $\vec{A}$ are proportional to input currents fed into these amplifiers. The M outputs (ψ_j) of the right-hand set of amplifiers represent constraint satisfaction. As indicated in the figure, the output (ψ_j) of the jth amplifier on the right-hand side injects current into the input lines of the V_i variable amplifiers by an amount proportional to $-D_{ji}$, the negative of the constraint coefficient for the ith variable in the jth constraint equation. Each of the M ψ_j amplifiers is fed a constant current proportional to the jth bound constant (B_j) and receives input from the ith variable amplifier by an amount proportional to D_{ji}. Like all of the amplifiers in Fig. 1, each of the V_i amplifiers in the linear programming network has an input capacitor C_i and an input resistor ρ_i in parallel, which connect the input line to ground. The input–output relations of the V_i amplifiers are linear and characterized by a linear function g_i in the relation $V_i = g(u_i)$. The ψ_i amplifiers have the nonlinear input–output relation characterized by the function

$$\psi_j = f(u_j), \qquad u_j = \vec{D}_j \cdot \vec{V} - B_j$$

where

$$\begin{aligned} f(z) &= 0, \qquad z \geq 0 \\ f(z) &= -z, \qquad z < 0. \end{aligned} \tag{16}$$

This function provides for the output of the ψ amplifiers to be a large positive value when the corresponding constraint equation it represents is being violated. (The specific form of $f(z)$ used here was chosen for convenience in building a corresponding real circuit and the stability proof only depends upon f being a function of the variable $z = \vec{D}_j \cdot \vec{V} - B_j$ (see below).) If we assume that the response time of the ψ_j is negligible compared to that of the variable amplifiers, then the circuit equation for the variable amplifiers can be written

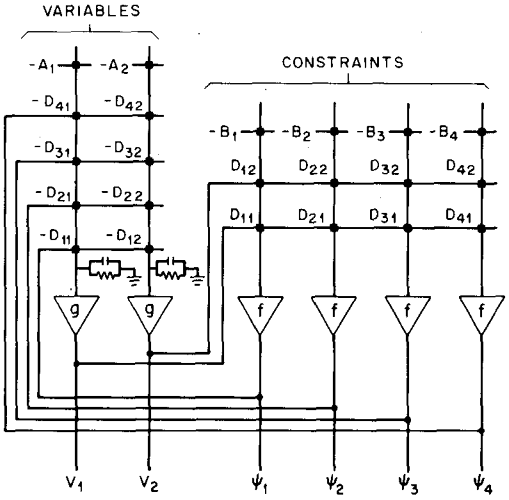


Fig. 8. The organization of a network which will solve a 2-variable 4-constant linear programming problem.

$$\begin{aligned} C_i \frac{du_i}{dt} &= -A_i - \frac{u_i}{R} - \sum_j D_{ji} f(u_j) \\ &= -A_i - \frac{u_i}{R} - \sum_j D_{ji} f\left(\vec{D}_j \cdot \vec{V} - B_j\right). \end{aligned} \tag{17}$$

Now consider an energy function of the form

$$E = (\vec{A} \cdot \vec{V}) + \sum_j F\left(\vec{D}_j \cdot \vec{V} - B_j\right) + \sum_i \frac{1}{R} \int_0^{V_i} g^{-1}(V)\, dV$$

where

$$f(z) = \frac{dF(z)}{dz}. \tag{18}$$

Then the time derivative of E is

$$\frac{dE}{dt} = \sum_i \frac{dV_i}{dt} \left[\frac{u_i}{R} + A_i + \sum_j D_{ji} f\left(\vec{D}_j \cdot \vec{V} - B_j\right) \right]. \tag{19}$$

But, substituting for the bracketed expression from the circuit equation of motion for the V_i amplifiers (17) gives

$$\frac{dE}{dt} = -\sum_i C_i \frac{dV_i}{dt} \frac{du_i}{dt} = -\sum_i C_i g^{-1}(V_i) \left(\frac{dV_i}{dt} \right)^2. \tag{20}$$

Since C_i is positive and $g^{-1}(V_i)$ is a monotone increasing function, this sum is nonnegative and

$$\frac{dE}{dt} \leq 0; \quad \frac{dE}{dt} = 0 \rightarrow \frac{dV_i}{dt} = 0, \qquad \text{for all } i. \tag{21}$$

Thus as for the network in Fig. 1, the time evolution of the system is a motion in state space which seeks out a minima to E and stops. The network in Fig. 8 should not show any oscillation even though there are nonsymmetric connection strengths between the two sets of V_i and ψ_j amplifiers, as long as the ψ_j are sufficiently fast.

A small computational network was constructed out of conventional electronic components to solve a 2-variable

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

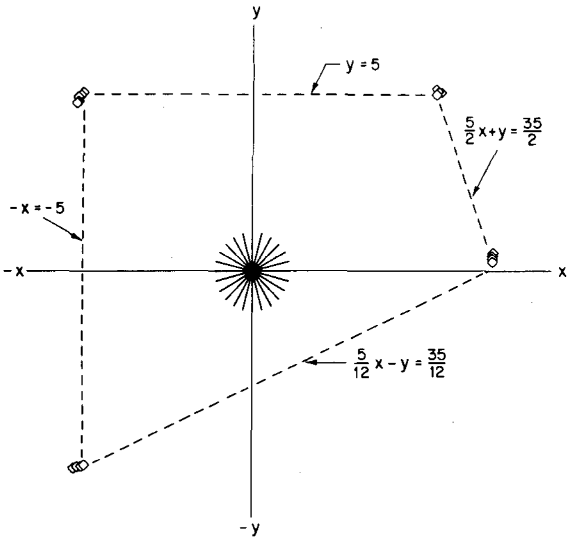


Fig. 9. A plot of the measured values of x and y for the linear programming network described in the text, as a function of the gradient of the optimization plane. The set of gradients is depicted by their projections onto the x, y plane, drawn as vectors from the origin.

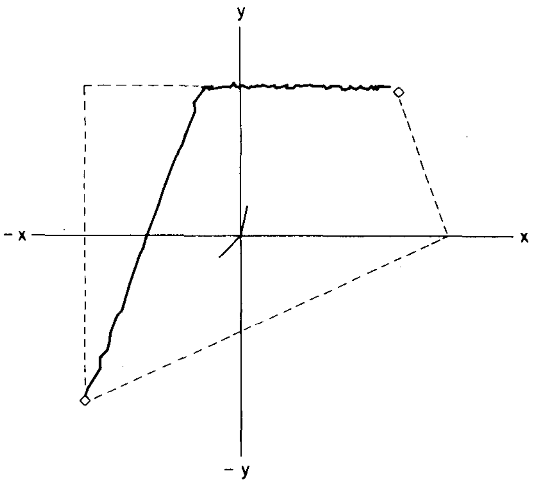


Fig. 10. The trajectory of x and y for the circuit described in the text as the gradient (indicated by the two vectors from the origin) is rapidly switched.

problem with four constraints using the network organization of Fig. 8. A simple op amp/diode active clamp circuit was used to provide the nonlinear f input–output function. The equations of constraint for the two variables (x and y) were

$$\begin{aligned} y &\leq 5 \\ -x &\leq 5 \\ \frac{5}{12}x - y &\leq \frac{35}{12} \\ \frac{5}{2}x + y &\leq \frac{35}{2}. \end{aligned} \tag{22}$$

These equations defined the connection strengths (D_{ji}) and the input currents (B_j) for the ψ_j amplifiers. In the xy plane characterizing the solution space, they defined the simplex shown in Fig. 9. A microcomputer-based data acquisition system was used to control the circuit and to measure the output voltages of the V_1 and V_2 amplifiers which corresponded to the x, y solutions, as a function of rapidly changing sets of input currents supplied to the input lines of these amplifiers. As indicated in Fig. 8, these input currents correspond to the coefficients A_i in the cost function which is to be minimized. For this simple 2-variable problem, the cost function can be geometrically thought of as a plane defined by the equation $z \doteq A_1x + A_2y$ hovering above the xy plane, and the direction of the gradient of that plane $A_1\hat{x} + A_2\hat{y}$ can be represented by a vector in the xy solution plane. The lowest point on the portion of this cost plane lying above the feasible solution space in the xy plane lies above the optimum simplex point. As the cost function is changed, the cost plane tilts in a new direction, the gradient projection in the xy plane rotates, and the optimum simplex point may also change. We recorded the values of x and y computed by the network for a set of cost functions. The operating points of the circuit are plotted in Fig. 9. Each diamond represents the location in the xy space at which the network stabilized at as the cost-plane gradient vector (indicated by the array of short line segments emanating from the origin) was swept in a circle. The circuit was stable at the optimum simplex points corresponding to the correct constrained choice for a given gradient direction.

In another experiment, the variable amplifiers were artificially slowed using large input capacitance and the trajectory followed by V_1 and V_2 was collected by rapid data sampling as the gradient was rapidly switched in direction. The trajectory is shown in Fig. 10. The network follows the gradient until it reaches a constraint wall which it then follows until the optimum simplex is reached. Since the solution space is always convex for linear programming problems, the network is guaranteed to find the optimum solution.

V. Conclusions

We have demonstrated how interconnected networks of simple analog processors can be used to solve decomposition/decision problems and linear programming problems. Networks for both problems were designed using conceptual tools which allow one to understand the influence of complicated feedback in highly interconnected networks of analog processors. There appears to be a large class of computation problems for which this simple concept of an "energy" function generates a complete stable circuit design without the need for a detailed dynamic analysis of stability. The function produces the required values of the many resistors from a short statement of the overall problem.

The two basic computations—digital decomposition and the linear programming network—are quite different computations in several respects. In the decomposition/decision networks discussed, the answers are digital, and this requirement that the stable states of the network lie on the corners of the solution space determines the highly nonlinear input–output relations for the variable amplifiers. Also, the equations of motion for the individual elements in the

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.

network *are of no intrinsic relevance* to the problem to be solved; they are a program which is used to compute the correct solution. In contrast, the amplifiers for the variables in the linear programming network are linear and furthermore the circuit equations of the linear programming network (17) have a more direct relationship to the problem to be solved; the constraint relationships are explicitly represented. This is similar to conventional methods of analog computation in which the processing elements are chosen to compute specific terms in a differential equation to be solved. In fact, a computational circuit similar to that in Fig. 8 has been described [7]. Here we have analytically shown the stability of this circuit design and illustrate it as a limiting case of more general networks for which the circuit equations do not necessarily relate to the problem to be solved. Another distinction is that the signal decision/deconvolution circuit makes a decision on the basis of the absolute values of its analog inputs, while the linear programming circuit decisions are based only on the relative values of the input amplitudes $\vec{A}$. This self-scaling property is often desired in signal processing and pattern recognition.

The practical usefulness of analog computational networks remains to be determined. Here, we have demonstrated, that for "simple" computational tasks and well-defined initial conditions, the networks can sometimes be guaranteed of finding the global optimum solution. The major advantage of these architectures is their potential combination of speed and computational power [1]. Interesting practical uses of such circuits for complicated problems necessitate huge numbers of connections (resistors) and amplifiers. Such circuits might be built in integrated circuit technology. Work has begun on questions of the microfabrication of extensive resistive connection matrices [8], [9]. Optical implementations of such circuits are also feasible [10].

References

[1] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions optimization problems," *Biological Cybern.*, vol. 52, pp. 141–152, 1985.

[2] J. J. Hopfield and D. W. Tank, "Collective computation with continuous variables," in *Disordered Systems and Biological Organization*, E. Bienenstock, F. Fogelman, and G. Weisbuch, Eds., Berlin, Germany: Springer-Verlag, 1985.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability.* New York: Freeman, 1979.

[4] S. E. Fahlman, "Three flavors of parallelism," in *Proc. of the Fourth National Conf. of the Canadian Society for Computational Studies of Intelligence*, Saskatoon, Sask., Canada, May 1982.

[5] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 81, pp. 3088–3092, 1984.

[6] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 79, pp. 2554–2558, 1982.

[7] I. B. Pyne, "Linear programming on an electronic analogue computer," *Trans. AIEE, Part I* (*Comm. & Elect.*), vol. 75, 1956.

[8] M. Sivilotti, M. Emerling, and C. Mead, "A novel associative memory implemented using collective computation," 1985 Conf. on VLSI's, H. Fuchs, Ed., Rockville, MD: Computer Science Press, 1985, p. 329.

[9] L. D. Jackel, R. E. Howard, H. P. Graf, R. Straughn, and J. Denker, "Artificial neural networks for computing," in *Proc. of the 29th Int. Symp. on Electron, Ion, and Photon Beams*, to be published in the *J. Vac. Sci. Tech.*.

[10] D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Opt. Lett.*, vol. 10, pp. 98–100, 1985.

David W. Tank was born in Cleveland, OH, on June 3, 1953. He received his undergraduate education from Case Western Reserve University, Cleveland, OH, and Hobart College, Geneva, NY. He received the Ph.D. degree in physics from Cornell University, Ithaca, NY in 1983.

From 1983 to 1984 he was a Post-Doctoral fellow at AT&T Bell Laboratories in Murray Hill, NJ. He has remained at Bell Laboratories, joining the Molecular Biophysics Research Department in 1984. His research interests concern the biophysics of individual nerve cells, neural representations and coding, and the computational properties of neural circuits.

John J. Hopfield received the B.A. degree from Swarthmore College in 1954 and the Ph.D. degree in physics from Cornell University, Ithaca, NY, in 1958.

His research has included work on electron transfer in photosynthesis, accuracy and proofreading in biomolecular synthesis, studies of "neural" networks in biological computation, and optical properties and impurity levels of semiconductors. He is currently Roscoe G. Dickonson Professor of Chemistry and Biology at the California Institute of Technology and a member of the Molecular Biophysics Research Department at AT&T Bell Laboratories, Murray Hill, NJ.

Authorized licensed use limited to: INDIAN INSTITUTE OF TECHNOLOGY KANPUR. Downloaded on December 09,2022 at 07:32:49 UTC from IEEE Xplore. Restrictions apply.