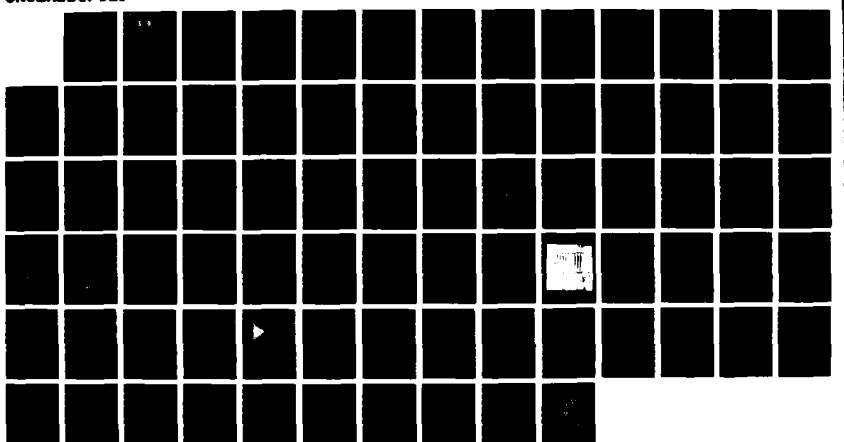
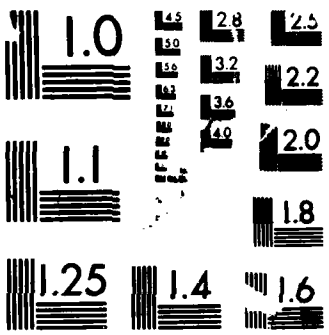
AD-A187 862 OPTICAL COMPUTING RESEARCH(U) STANFORD UNIV CA STANFORD 1/1
ELECTRONICS LABS J W GOODMAN 30 OCT 87
AFOSR-TR-87-1635 AFOSR-86-0203

UNCLASSIFIED F/G 20/6 NL





MICROCOPY RESOLUTION TEST CHART
DS-1963-A

DTIC
UNCLASSIFIED
SECURITY CLASSIFI

AD-A187 862

(2)

REPORT DOCUMENTATION PAGE

DTIC ELECTE NOV 17 1987 S D

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY UNCLASSIFIED	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CD	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 87-1635
6a. NAME OF PERFORMING ORGANIZATION STANFORD UNIVERSITY ELECTRONICS LABORATORIES	6b. OFFICE SYMBOL (If applicable)
6c. ADDRESS (City, State and ZIP Code) STANFORD, CALIFORNIA 94305	7a. NAME OF MONITORING ORGANIZATION Same as 8a
	7b. ADDRESS (City, State and ZIP Code) Same as 8b
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NE
	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-86-0283
8c. ADDRESS (City, State and ZIP Code) Building 410 Bolling Air Force Base Washington, DC 20332-6448	10. SOURCE OF FUNDING NOS. PROGRAM ELEMENT NO.: 61102F PROJECT NO.: 2305 TASK NO.: B4 WORK UNIT NO.:
11. TITLE (Include Security Classification) UNCLASSIFIED OPTICAL COMPUTING RESEARCH	
12. PERSONAL AUTHOR(S) Professor Joseph W. Goodman	
13a. TYPE OF REPORT FINAL	13b. TIME COVERED FROM 9/30/86 TO 9/30/87
14. DATE OF REPORT (Yr., Mo., Day) 87/10/30	15. PAGE COUNT 76
16. SUPPLEMENTARY NOTATION	

17. COSATI CODES		
FIELD	GROUP	SUB. GR.

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Work Accomplished: OPTICAL INTERCONNECTIONS - the powerful interconnect abilities of optical beams have led much optimism about the possible roles for optics in solving interconnect problems at various levels of computer architecture. Examined were the powerful requirements of optical interconnects at the gate-to-gate and chip-to-chip levels. OPTICAL NEUTRAL NETWORKS - basic studies of the convergence properties of the Holfield model, based on mathematical approach - graph theory. OPTICS AND ARTIFICIAL INTELLIGENCE - review the field of optical processing and artificial intelligence, with the aim of finding areas that might be particularly attractive for future investigation(s).

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr. C.L. Giles Professor Goodman, Joseph W.	22b TELEPHONE NUMBER (Include Area Code) (202) (415) 723-2883 767-4931
22c. OFFICE SYMBOL UNCLASSIFIED NE	

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

SECURITY CLASSIFICATION OF THIS PAGE

OPTICAL COMPUTING RESEARCH

FINAL REPORT

on AFOSR Grant 86-0283

Submitted to the Air Force Office of Scientific Research

Joseph W. Goodman, P.I.
Stanford University

~~September~~ Oct. 30, 1987

ABSTRACT

This report summarizes the work accomplished since September 1, 1986 on Air Force Office of Scientific Research Grant 86-0283. Work has been in progress in three different areas: 1) Optical interconnections; 2) Neural networks; and 3) Optics and artificial intelligence. Various administrative matters pertinent to the grant are also discussed.

I. INTRODUCTION

This document contains a summary of the work accomplished under Grant AFOSR 83-0166 during the time period September 1, 1986 through September 30, 1987. Section II contains a summary of the work accomplished. This summary is supplemented by appendices. Section III is devoted to various administrative matters pertinent to the grant.

II. WORK ACCOMPLISHED

(a) Optical Interconnections

The powerful interconnect abilities of optical beams have led to much optimism about the possible roles for optics in solving interconnect problems at various levels of computer architecture. It is already well established that fiber optics will play an important role in enabling high-speed machine-to-machine communication. At lower levels of architecture, e.g. module-to-module within a machine, board-to-board within a module, chip-to-chip on a board, and gate-to-gate on a chip, the proper role for optical interconnections is not as well established. One of the most important unanswered questions facing the field of optical interconnects is how far down this hierarchy of levels of interconnects optics will play a useful role.

We have examined the power requirements of optical interconnects at the gate-to-gate and chip-to-chip levels. Our findings can be summarized roughly as follows. At the level of gate-to-gate interconnects, optical interconnects require substantially more power than electrical interconnects. The extra power required of an optical interconnect stems from two sources. First, the conversion of electrons to photons in a very good laser diode has about a 20-25 % overall power efficiency (i.e. approximately one photon is generated for every four or five electrons). Second, the lowest threshold laser diodes reported to date require approximately one milliampere of current to reach threshold and have a bandgap of

about 1.4 volts. Thus for any interconnect that can be accomplished electrically with less than one milliwatt, there is a power disadvantage to optics. Typically at the gate-to-gate level, the anticipated requirements to continuously charge and discharge an interconnect line plus the input capacity of the destination gate to a reasonable threshold voltage (say 1 volt) in a time of, for example, one nanosecond are considerably smaller than a milliampere. However, at the chip-to-chip level, the conclusions are quite different. If the electrical line is short enough and the bandwidth is low enough that the line can be unterminated, then the electrical problem is primarily that of charging the relatively large capacitances of bonding pads on the source and destination chips. In this case the electrical power required of the interconnect is comparable with that required of an optical interconnect line. If the interconnect length and bandwidth lead to the requirement that the line be terminated, then the power dissipation associated with the electrical line increases dramatically, and optics has a distinct power advantage with respect to electronics. This advantage is even more dramatic if the interconnect line has significant fan-out. See a more detailed discussion of power issues in Appendix A, which is a preprint of a chapter on optical interconnects we have written for a book on optical computing being edited by H.H. Arsenault.

A major conclusion is that a sufficient (but not necessary) condition for interest in optical interconnects will be present whenever an electrical interconnect line must be terminated by its characteristic impedance. The condition is not a necessary one because it does not address the issues of mutual coupling, EMI, space requirements, and ground loop elimination, any one of which might of itself lead to optics being the solution of choice, regardless of any power disadvantage that might hold.

The calculations and assumptions behind these conclusions are presented in the Ph.D. thesis of Raymond Kostuk, published in the Autumn of 1986 and available on request. We are not attaching the thesis as an appendix to this report because of its large size (305 pages). The conclusions will be brought to the attention of the optics communities through

the book chapter mentioned above and a chapter on optical interconnects now being written by Dr. Kostuk for another book on optical computing.

The thesis of Raymond Kostuk also contained a large body of material on holographic optical elements, as they pertain to the interconnect problem. Two papers by us have appeared in *Applied Optics* on this subject , as discussed in Section III in more detail.

(b) Optical Neural Networks

The work undertaken jointly with Prof. Mitsuo Takeda of the University of Electro-communication in Japan on computation using neural networks was published in *Applied Optics* during the early months of this grant year. A reprint of this paper is attached as Appendix B. Since that time our work has focused on two different aspects of the neural computing problem. First, we have undertaken some basic studies of the convergence properties of the Hopfield model, based on a novel mathematical approach - graph theory. Each neuron can be regarded as a node in a directed graph, and each interconnection between two neurons can be regarded as a weighted edge in the graph. It can be shown that the Hopfield network in effect performs a local search for a minimum cut through the graph. This is but one example of the new insights that can be gained from bringing the power of graph theory to bear on neural network problems. The theory so developed is applicable to both optical and non-optical neural networks. Appendix C contains a reprint of a paper on our work that will appear in the *Proceedings of the First International Conference on Neural Networks* (San Diego, June, 1987). Appendix D contains a preprint of our paper to be presented at the IEEE Conference on Neural Information Processing Systems, to be held in Boulder, Colorado, in November, 1987. An extended version of Appendix C has been accepted for publication in the *IEEE Transactions on Information Theory* after minor modifications.

A second area of study in the neural network area is more practical and problem-oriented in its nature. Recognizing that neural networks are best suited for so-called random problems, we have been examining the problem of signature verification as one to which neural net ideas might profitably be applied. No individual signs his signature twice in exactly the same way. Therefore it is impossible to store all versions of an individual's signature, nor is it possible to store all the forgeries of that signature. The approach under investigation, therefore, is to train a neural net with correct and incorrect signatures, and to assess its performance in identifying forgeries. A three level neural network is being simulated, with a layer of hidden neurons. Important questions to be answered include the dependence of net performance on the number of neurons used, the number of hidden layers used, and the number of signatures used in training. The signature verification problem is one of particular interest from the point-of-view of optical neural nets, because the inputs are inherently in optical form.

Effective January 1, 1987, the simulations involved in the signature verification work were transferred to other funding.

(c) Optics and Artificial Intelligence

During the current grant year we initiated a new effort which reviewed the field of optical processing and artificial intelligence, with the aim of finding areas that might be particularly attractive for future investigation. This study was undertaken at no cost to the grant, since the student involved had other support for the duration of this academic year. The summary examined work under way at the University of Colorado, Carnegie Mellon University, Honeywell, Johns Hopkins University, the University of Southern California, and the BDM corporation. This ongoing work in optical approaches to AI was reviewed in order to identify what we believe are the most promising areas to investigate in this field. The conclusion of this investigation is summarized as follows. We believe the most

promising and unique avenue for investigation in this area is an attempt to perform predicate calculus optically, possibly using matrix-vector multipliers. Predicate calculus is by far the most widely used artificial intelligence tool in current AI software systems. Furthermore, to the best of our knowledge, optical implementations of predicate calculus are not now being investigated in other laboratories. Work at the University of Colorado is focusing on optical implementations of propositional calculus, while work at Carnegie Mellon University is aimed at using matrix-vector multipliers for associative retrieval. The work we propose is different from of both of these efforts. We hope to pursue this issue beyond the present stage in a new grant from AFOSR.

III. ADMINISTRATIVE MATTERS

(a) Publications

Publications submitted or appearing on work fully or substantially supported by the grant during the past year are as follows:

1. M. Takeda and J.W. Goodman, "Neural networks and computing: number representations and programming complexity", *Applied Optics*, Vol. 25, pp. 3033-3046 (1986).

2. P. Idell and J.W. Goodman, "Design of optical imaging concentrators for partially coherent light: absolute encircled energy criterion", *J. Opt. Soc. Am. A*, Vol. 3, pp. 942-953 (1986).

3. R.K. Kostuk, J.W. Goodman, and L. Hesselink, "Volume reflection holograms with multiple gratings: an experimental evaluation", *Applied Optics*, Vol. 25, pp. 4362-4369 (1986).

4. M. Nazarathy and J.W. Goodman, "Systolic lattice processing by optical coupled-mode device arrays", *Optical Engineering*, Vol. 26, No. 3, pp. 256-263 (1987).

5. R.K. Kostuk, J.W. Goodman, and L. Hesselink, "Design considerations for holographic optical interconnections", Applied Optics (Accepted for publication).

6. R.K. Kostuk, J.W. Goodman, L. Hesselink, "Volume reflection holograms with multiple gratings: an experimental and theoretical evaluation", Applied Optics, Vol. 25, pp. 4362-4369 (1986).

7. J. Bruck and J.W. Goodman, " A generalized convergence theorem for neural networks and its applications in combinatorial optimization", Proceedings of the First International Conference on Neural Networks (in press).

8. J. Bruck and J.W. Goodman, A generalized convergence theorem for neural networks and its application to combinatorial optimization", submitted to IEEE Trans. on Info Theory (1987). Accepted for publication after revision.

9. J. Bruck and J.W. Goodman, "On the power of neural networks for solving hard problems", to appear in the *Proceedings of the IEEE Conference on Neural Information Processing Systems*, Boulder, Colorado, November 1987.

10. J.W. Goodman, "Optics as an Interconnect Technology", to be published in 1988 as a chapter of a book on optical computing, edited by H.H. Arsenault.

(b) Presentations

Oral presentations on work supported by the grant include the following:

1. R.K. Kostuk, "Design considerations for holographic optical interconnections", presented at the Annual Meeting of the Optical Society of America, Seattle, Washington, October 1986.

2. J.W. Goodman. "Optical computing: an overview", presented at the OSA Topical Meeting on Optical Computing, Incline Village, Nevada, March 1987 (invited paper).

3. J.W. Goodman, "Optical communications and computing", presented at the biannual meeting of the Israel Section of the IEEE, Tel Aviv, Israel, April 1987 (plenary paper).

4. J.W. Goodman, "Optics as an interconnect technology", Workshop on GaAs on Si, Marina-del-Rey, CA, June 1987

5. J. Bruck , " A generalized convergence theorem for neural networks and its applications in combinatorial optimization", First International Conference on Neural Networks, San Diego, CA June 1987.

(c) Students Supported by the Grant

The students supported by the grant during part or all the past year were Jehosua Bruck, Raymond Kostuk, and Dorothy Mighell. R. Kostuk received the Ph.D. in late 1986 and is now a member of the faculty at the Optical Sciences Center of the University of Arizona.

(d) Honors received by the Principle Investigator

During the past grant year, Dr. Goodman received the following honors:

1. Recipient, 1987 IEEE Education Medal

2. Elected Member of the National Academy of Engineering

3. Recipient of the 1987 Dennis Gabor Award of the SPIE

4. Elected President, International Commission for Optics.

A

OPTICS AS AN INTERCONNECT TECHNOLOGY

Joseph W. Goodman

Department of Electrical Engineering

Stanford University

Stanford, California 94305

1. Introduction

The hardware portion of a digital computing system can be regarded in most general terms as a collection of many nonlinear elements within which signals must interact (the gates), together with interconnections between those elements, or between groups of such elements. The groups of elements can be of various sizes and complexities, depending on the level of architecture of concern. The function of the interconnections may be to communicate information to or from processing subunits, memories subunits, or users,or to transfer control signals or program segments to hardware subunits of various kinds.

The variety of different kinds of interconnect problems can be appreciated in the context of a listing of several levels of computer architecture within which interconnections play a fundamental role. Starting at the highest levels of architecture and working downward to lower levels, we have:

Machine-to-machine interconnections.. The interconnections are required to transfer messages of various kinds, including electronic mail, files, and

information from shared databases. The distances involved typically vary from several meters to a few kilometers.

Processor-to-processor interconnections. In a multiprocessor environment within a single machine, interconnections are required between different processors, and between processors and certain shared resources, such as memory. In many cases it is necessary to change the interconnect pattern dynamically in time. The distances involved may vary from as little as a few centimeters to a few meters.

Board-to-board interconnections. Within a single processing unit, there usually exist several electronic boards. These boards must interchange information, and usually do so by means of some form of data bus. The distances involved can vary from a very few centimeters to perhaps as much as one meter.

Chip-to-chip interconnections. On a single board there typically exists a multitude of integrated circuit chips, many of which must communicate with one another. The communication distances involved range from of the order of 0.1 centimeter to as much as a few tens of centimeters. A special case is that of wafer-level interconnection, in which various chips on a single wafer must communicate.

Intra-chip interconnections. A single integrated circuit chip typically contains thousands of interconnections between gates and between different functional subunits of the chip. In addition, a substantial system of interconnections exists between the chip itself and the pins that connect it to

the outside world. This intrachip level we regard as the lowest level of the interconnection hierarchy. The distances involved range from a few micrometers to at most a very few centimeters.

An examination of the above hierarchy of interconnect problems reveals that optics is now penetrating the highest levels. Machine-to-machine communication via optical fibers is now a commercial reality, and serious attempts are in progress to bring optics to the next lower level of architecture, processor-to-processor interconnection within a single machine. Research is also under way at the chip-to-chip level. How far down this hierarchy of interconnection problems optics will eventually penetrate is a subject for speculation.

It is the goal of this chapter to examine the properties of optical signals that make them attractive as an interconnect technology. Some speculation as to possible future developments will also be included. Several other discussions of this subject are available in the literature[1,2,3,4].

2. Why Use Optics For Interconnections?

The physical properties desired of an interconnect technology are markedly different (and indeed in many respects quite the opposite of) the physical properties required of a gate technology. An interconnect technology should ideally have the property that interactions between different interconnections are minimum or non-existent. Thus signals flowing through one interconnection should not couple to or otherwise influence signals flowing in another interconnection. There are fundamental differences between electrons and photons that are pertinent in this regard. Several points-of-view are possible in explaining these differences. From the most basic perspective, electrons are members of the class of particles known as fermions, while photons belong to the class known as bosons.

According to the Pauli exclusion principal, no two fermions can occupy the same cell of phase space, whereas any number of bosons can share a common cell. This fact can be viewed as implying that electrons must fundamentally suffer mutual interactions that prevent violation of the exclusion principal, while no such interactions need exist for photons.

A somewhat more straightforward point-of-view rests on the fact that electrons are charged particles while photons carry no charge. Moving electrons thus generate stray electric and magnetic fields, which in turn couple signals into proximate conducting lines. No such fundamental coupling mechanism exists for streams of photons, although from the practical point-of-view, some level of coupling can arise through optical scattering if care is not taken.

Regardless of the point-of-view, there is a fundamental conclusion that emerges from the discussion: *optical interconnections potentially offer a freedom from mutual coupling effects not afforded by conventional electronic interconnects.* This potential advantage of optics becomes more and more important as the bandwidth of the desired interconnections increases, for the strength of mutual coupling associated with electrical interconnects is proportional to the frequency of the signals propagating on the interconnect lines.

A second potential advantage of optical interconnections is an extra flexibility of routing. Electrical interconnect paths can not cross, and therefore must be routed over or under one another through multiple interconnect layers. Optical interconnections can indeed be routed through one another without any deleterious effects. Unlike electrical interconnect paths, which must reside near a ground plane to assure that stray electric fields are properly terminated, optical interconnects need not remain near a ground plane, and indeed can be routed in a flexible manner through three-dimensional space.

A third advantage of optical interconnects rests on a partial freedom from certain capacitive loading effects. For an electrical interconnection, the delivery of signals to a number of different devices or subunits of a system requires that the interconnect line drive a total capacitance consisting of the capacitance of the interconnect line plus the capacitances of all the devices or subsystems attached to that line. The capacitance of the line itself is proportional to the length of the interconnection. For both optical and electrical interconnects, the basic signal-carrying streams (comprised of photons and electrons, respectively) must be divided between the various termination points where information is to be delivered. However, in the electrical case, if the connection is long enough, a significant number of electrons are diverted to charging the capacitance of the interconnect line, and are therefore not available for charging the capacitances of the devices at the termination points. No such line charging phenomenon is present for an optical interconnection, although the equivalent of a resistive loss is present if the optical interconnect line has significant absorption or scattering, or if the electrical-to-optical or optical-to-electrical converters (sources and detectors, respectively) have low quantum efficiency.

A fourth possible advantage of optics rests on its potential for supplying dynamically changeable interconnect devices. Since photon-based interconnects require no mechanical contacts, interconnect re-routing can be accomplished simply by changing directions of optical beams. While much work remains to realize dynamic routing elements with interestingly large numbers of connections and speeds of reconfiguration, nonetheless the potential for optics in this role is intriguing [5,6].

3. Types of Optical Interconnections

An optical interconnection performs the task of delivering modulated light generated at a source to a detector where the modulation is recovered. The interconnection should be efficient in that as many as possible of the available photons should be delivered to the desired destination. In addition, the interconnection should be as free as possible from dispersion that might limit the bandwidth of the modulation recoverable at the receiver. There are in fact a variety of optical methods that could be used as the basis for realization of optical interconnections. In this section we briefly describe the various possibilities.

The first method for realizing optical interconnections that comes to mind is by means of *optical fibers*, as indicated conceptually in Fig. 1. Fiber-optic technology is having

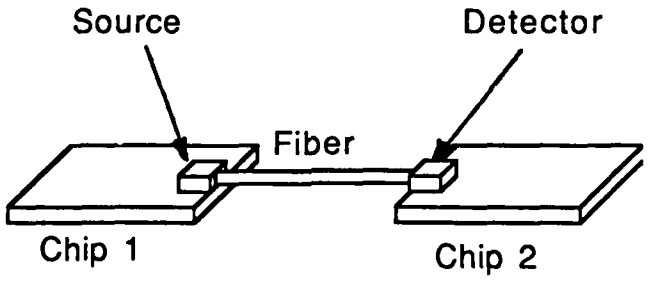


Figure 1. Chip-to-chip interconnect with optical fibers

enormous impact on telephone and data communications, particularly over distances of several to many kilometers. Commercial availability of fiber-optic networks is also beginning to be seen, used for connecting a number of digital computers with one another and with shared peripherals. It seems natural therefore to consider the possible use of fibers at other high levels of computer architecture. Optical fibers have many of the properties we might desire of an interconnect technology. They efficiently deliver photons coupled into the input to detectors at the output. Their losses are so low at lengths of a kilometer and less

(the distances of main interest here) that attenuation by the fibers themselves can be neglected. In addition, dispersion over such short distances is usually negligible at bandwidths of current interest. The above comments apply for both single-mode and multi-mode fibers, so there seems little motivation to utilize the more complicated and expensive single-mode technology in these applications when multi-mode solutions should be perfectly adequate. However, optical fibers are not necessarily the ideal solution for interconnect problems at all levels. In particular, at the lowest levels, i.e. intrachip and very nearby chip-to-chip interconnections, the problems of bending and looping fibers become severe, due to radiation losses induced by bending. For such problems, it might be argued that fibers are too much like wires, requiring a material path for the interconnection between every two points, and rather inflexible paths at that.

An alternative approach that may be applicable to intrachip communications is the use of *integrated optic* technology, illustrated in Fig. 2. This approach rests on the use of

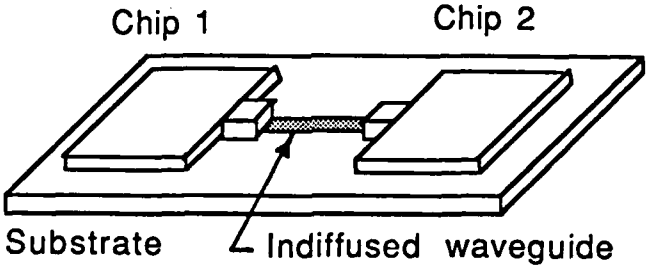


Figure 2 Chip-to-chip interconnect using integrated-optic waveguides

waveguides that are integrated in a planar substrate. Most common is the creation of waveguides in lithium niobate by indiffusion of titanium channels, but similar guides can be made by sputtering glass on SiO_2. While the losses associated with such waveguides are orders of magnitude higher than those associated with optical fibers, nonetheless the distances for intrachip communication are so short that these losses may not be of great

importance. However, losses, together with the problems associated with realizing large integrated-optic substrates, appear to rule out the use of integrated optics for levels of interconnection higher than intrachip or very nearby interchip.

An important practical problem with the integrated-optic approach is the coupling of light into and out of the waveguides. Butt coupling of light into such waveguides is common, and applicable when the source is a discrete device. Likewise end-to-end juxtaposition of a discrete detector and the waveguide output can serve to deliver the optical signals to the photosensitive surface of the detector.

In some applications, it may be desirable to place a passive waveguide substrate over an active integrated circuit, in which detectors and/or sources have been integrated. The problem of efficient coupling into and out of the waveguides is more difficult in this case, requiring the use of prism or grating couplers.

Of the various imaginable ways for using optics for interconnects,perhaps the simplest method from the conceptual point-of-view is that of *free-space unfocused broadcast*, shown in Fig. 3. For this method, a modulated optical signal, generated, for example, by a

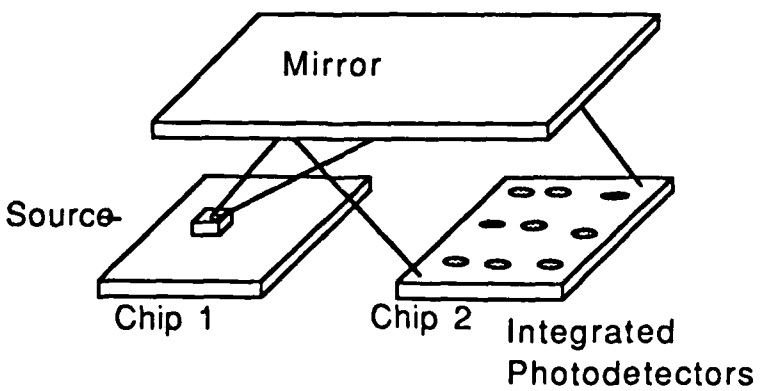


Figure 3. Free-space unfocused broadcast

laser diode, is transmitted in a broad and unfocused beam, portions of which fall upon one or more detectors. If the interconnection is from one source to a single receiver, then the connection is one-to-one. On the other hand, if the same signal must be delivered simultaneously to several receiving points, then the interconnection is said to have *fan-out* and to be one-to-many. The chief drawback of this approach to interconnection is the very low degree of efficiency with which photons are utilized. The fact that the light has been spread over a large area implies that only a small fraction of the optical power will be intercepted by any one detector. Since in most applications the electrons generated by photons at the detector must charge a capacitance on a gate, loss of photons implies that longer integration times will be required for the gate threshold voltage to be reached. Thus the speed capability of the circuit will be less than it could be with more efficient delivery of photons. A further disadvantage of this approach in many applications is the lack of parallelism in communication capability. Since all detectors receive all the signals transmitted by any one source, the communication channel realized by this approach must be time-shared. Only one source can be active at one time, thus eliminating all potential for parallelism. In spite of the drawbacks of this type of interconnection, it has been used in at least one experimental computer system as the basis for a common bus [7].

The final approach to be discussed can be called *free-space focused interconnection*, or more simply, *imaging interconnections*., as illustrated in Fig. 4. This method differs from that discussed above in that rather general optical focusing elements are used to place nearly all the available light onto the detector sites where it is required. The focusing elements generally must be realized by means of holography, and are referred to as holographic optical elements. Using such elements, a single source can be imaged onto one or more

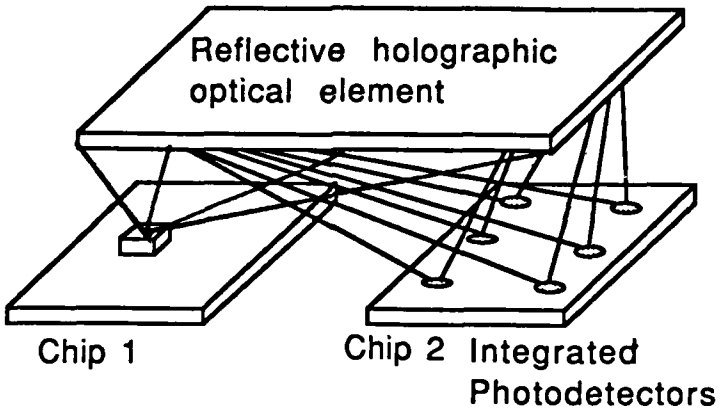


Figure 4. Imaging interconnections.

photodetectors with high efficiency, eliminating much of the waste of photons present in the unfocused case. The practical limits to efficiency depend on the amount of fan-out required, the material from which the hologram is made, and the geometry required. With the use of dichromated gelatin as a recording material, simple reflection holograms capable of imaging one source onto one detector using visible light can be made with efficiencies in excess of 99%. Less is known about making good holographic optical elements that work in the near infrared, where high-speed optical communication technology is prevalent (a recent reference is available on this subject [8]). Note that the use of of imaging interconnections reserves the possibility of parallelism in the interconnect network. One source can be imaged onto one photodetector, while another independent source is imaged onto another different photodetector. Each such channel can then operate independently. If holographic optical elements are to be used, it should be noted that the Bragg selectivity of a thick *transmission* grating is far superior to that of a thick *reflection* grating. Therefore, when several parallel interconnects are to be made independently, a transmission geometry is preferable to a reflection geometry.

4. Some Specific Properties of Optical Interconnections

In a previous section, we have discussed in a general way the differences that exist between optical and electronic interconnections, as well as the properties of optics that make it attractive as the basis for an interconnect technology. In this section we will discuss some more specific properties of optical interconnections, particularly properties that strongly influence the practicality of such techniques in real applications.

If optical interconnects are to be used to interconnect two electronic subunits at some level of architecture, then signals that are originally in electronic form must be converted to optical form, and following reception at the destination, they must be converted back into electronic form. Each conversion step has a finite efficiency, and it is important to quantify the impacts of the associated losses. It should be kept in mind that in most applications the termination of the interconnect link will be a capacitive load that must be charged to the threshold voltage of the logic devices that follow. To reach the threshold voltage, a certain minimum amount of charge must be deposited at the terminating device.

A flow of electrons over an electronic connection must be mimicked by a flow of photons over an optical connection. Thus a flow of current in the electronic case is analogously replaced by a flow of power in the optical case. It is, of course, necessary to take into account the finite efficiency of the optical source and the finite quantum efficiency of the optical detector in making any comparison.

Detector efficiency

In the case of a communication link at 800 nm wavelength, we can assume a 80% quantum efficiency for a silicon p-i-n photodiode, corresponding to a responsivity of approximately

0.5 amps/watt. Avalanche photodiodes have inherent gain, and as a consequence they are better converters of optical power into electrical current. Typical values of responsivity for such detectors are 50 to 100 Amps/Watt. However, we intentionally exclude avalanche photodiodes from consideration here, for two reasons. First, and foremost, we wish to compare optical and electronic interconnections without any gain mechanism present. Obviously both types of interconnections can be followed by devices or circuits with gain. Comparison of the two technologies then becomes a strong function of the detailed character of the gain mechanism, opening a Pandora's box of details that are beyond our central purpose here. Secondly, once gain is allowed at the end of the interconnection, the total power requirements no longer arise only at the transmitting end of the link. Rather, power is supplied at the receiving end as well, further complicating the comparison. It should also be mentioned that avalanche photodiodes require higher voltages than p-i-n diodes, and have greater temperature sensitivities, making their use in practice more complex than p-i-n diodes.

Source efficiency

The efficiency considerations discussed above apply only to photon-to-electron conversion at the receiving end of the interconnection. Equally important are the efficiency considerations for the initial electron-to-photon conversion at the transmitting end of the link.

Efficiencies in the range of 1% to 2% are typical of LED's operating in the near infrared. Such low efficiencies make it difficult for an optical interconnect to compete on a power basis with an electrical link. Therefore higher efficiency laser sources are of greater interest.

High efficiencies are realized in laser diodes only well above lasing threshold. Below threshold, a semiconductor laser source behaves essentially as an LED, and as such exhibits a relatively poor overall power efficiency. When using a laser diode, there exists a certain minimum amount of electrical power required to bring the source to threshold, and that electrical power is essentially wasted, since it generates very few photons. Any comparison of electrical and optical interconnects must ultimately take account of this minimum required threshold current. When an electrical interconnect requires less power than that needed to bring a laser diode to threshold, optics can not compete with the electronic solution. On the other hand, when the interconnect is characterized by a high degree of fan-out, the inefficiency associated with the finite laser threshold can be amortized over a large number of connections, making the optical interconnect more attractive as a solution.

As a typical state-of-the-art laser diode attractive for interconnect applications, we consider the GaAs single quantum well laser recently described in the literature[9]. The threshold current for this laser is just under 1 mA, and the required applied voltage is 1.4 volts. Thus to reach threshold a commitment of 1.4 mW of electrical power is required. In the middle of the lasing range, 5 mA of current are required to produce approximately 1.5 mW of output optical power. Thus the efficiency of the device in converting electrical to optical power is 21%. Efficiencies in the range of 20 to 25% are typical of good semiconductor lasers with outputs in the range of a few mW. Higher power semiconductor lasers can achieve higher efficiencies, even in excess of 50%. In our considerations, we will use a 25% efficiency number.

Interconnect efficiency

Finally, considerations of efficiency can not ignore the losses associated with the optical path from transmitter to receiver. These losses vary dramatically, depending on which of the various schemes discussed in the previous section are chosen for the optical interconnect. Most efficient would be a fiber-optic link, which for the geometries to which such technology can be applied and for the short distances of interest here, will suffer primarily from coupling losses at the input and output of the fiber. Losses of only a db or two should be possible. Least efficient will be the free-space broadcast schemes, for which the losses could easily reach 60 db in some applications (broadcast to a 10 μm x 10 μm detector on a 1 cm x 1 cm chip). Holographic distribution systems exhibit losses of 3 db to 10 db for bleached silver-halide-based reflection holograms, and considerably less loss for dichromated-gelatin-based reflection elements. Larger losses can be expected if the light from the optical source overfills the holographic optical element. Such would be the case for an LED source, but not necessarily for a semiconductor laser source.

5. Power Requirements for Optical Interconnections [10,11]

An important consideration for any interconnect line is the power required to drive that line, and the devices attached to it, at a given bit rate. It is instructive to compare conventional electronic interconnections with optical interconnections from this point-of-view, for some fundamental conclusions result. In the electrical case, the power requirements depend on whether the interconnect line is terminated or unterminated. Termination of a line is required if reflections from the end of the line pose a problem; it results in a substantially greater amount of power dissipation than if no termination is used. For an unterminated line a major portion of the power requirement arises from the need to charge the capacitance of the line and the capacitance of the device or devices attached to the line. Such power is

entirely reactive in nature. This conclusion changes when the bit rates become high enough, due to skin-effect losses in the conductors.

In the optical case, it is assumed that the same device that were driven by the electrical interconnect line are now driven by the output of the detector at the end of the line. The capacitance of those devices is unchanged, but the major power requirement now stems from the inefficiencies of the electron-to-photon and photon-to-electron conversion processes. These inefficiencies constitute real power loss, and therefore the drive power requirements are no longer reactive.

A rather simple example is revealing. Consider first the case of an electrical interconnection. Suppose that the electrical interconnect line is unterminated, and skin-effect losses are negligible. The drive power requirements in the electrical case then consist primarily of the reactive power needed to charge to the threshold logic voltage the capacitance of the interconnect line itself and the device capacitances attached to the line. The reactive power P required to charge a capacitance C to a voltage threshold level V in a fixed time τ is

$$P = \frac{CV^2}{2\tau}. \qquad (1)$$

Hence for a fixed voltage threshold level and a fixed charging time, the required power is directly proportional to the capacitance that must be charged. Let the capacitance of the device attached to both of the lines be represented by C_d and the capacitance of the electrical interconnect line itself be C_1. The reactive power P_e required for the electrical interconnect line is then

$$P_e = \frac{(C_l + C_d)V^2}{2\tau}. \qquad (2)$$

Consider next the case of an equivalent optical interconnection. A real electrical power P_{eo} must be supplied to the optical source in order to ultimately charge the same device capacitance, as well as the detector capacitance C_d. We calculate the required P_{eo} in the following manner. First, since by definition capacitance is the amount of charge stored for a given applied voltage, the charge required at the end of the interconnection in order to bring the detector capacitance and device capacitance to voltage V is

$$Q = V\,(C_D + C_d). \qquad (3)$$

To deposit that charge in time τ requires a current

$$i = \frac{V(\,C_D + C_d\,)}{\tau}. \qquad (4)$$

Let the responsivity of the detector be $\mathcal{R}$, in which case the optical power required to generate the current above is .

$$P_o = \frac{V(\,C_D + C_d\,)}{\mathcal{R}\tau}. \qquad (5)$$

Finally, taking account of the finite power efficiency η_s of the source, the total electrical power required to drive the optical link is

$$P_{eo} = \frac{V(\,C_D + C_d\,)}{\eta_s \mathcal{R}\tau}. \qquad (6)$$

Note that different methods were required to calculate P_e and P_{eo} due to the fact that in the former case we were dealing with a constant voltage source applied to the line, whereas in the optical case the capacitances are charged by a constant current source.

It is now possible to compare the powers required of the two technologies. The power required of the optical link will be less than the power required of the electrical link when $P_{eo} < P_e$; i.e. when

$$\frac{C_D + C_d}{\eta_s \mathcal{R}} < \frac{(C_l + C_d)V}{2}, \tag{7}$$

or equivalently when the electrical line capacitance satisfies

$$C_l > 2\frac{C_D + C_d}{V\eta_s \mathcal{R}} - C_d. \tag{8}$$

If we take the detector responsivity to be 0.5 watts/amp, and the laser efficiency to be 25%, the optical link will be superior to the electrical link whenever the electrical line capacitance satisfies

$$C_l > \frac{16}{V}(C_D + C_d) - C_d. \tag{9}$$

Note that the higher the threshold voltage required by the devices, the more favorable the optical link becomes. Finally it should be noted that in the chip-to-chip communication problem, the electrical interconnection is usually accomplished via bonding pads at each end of the line. The capacitances of the bonding pads are very large compared with the capacitances of a gate, and hence in this case the bonding pad capacitances should replace the gate capacitance when calculating the required electrical power.

The power requirement comparison above is, in a rather hidden way, intrinsically connected with the issue of the relative isolation of two adjacent interconnections. For the case of an electrical interconnection, a high degree of isolation can be achieved if the conducting lines are kept very close to a ground plane. Thus if the conducting line is separated from a ground plane by a thin dielectric layer, or if it is sandwiched between two ground planes with thin dielectric layers providing separation, the isolation of the line dramatically improves. However, accompanying this increased isolation there comes fundamentally an increase in the line capacitance, and therefore a greater power requirement for the electrical interconnection. Improved isolation comes from the more effective termination of electric field lines on the conducting ground plane or planes, resulting in less stray capacitance between lines. In turn, the increased number of field lines terminating on the ground plane implies that the line itself has a greater ability to store charge when a fixed voltage is applied, and therefore a greater inherent capacitance. Thus there is a direct tradeoff between isolation between lines and the electrical power required to drive those lines. No such tradeoff exists in the case of optical interconnections. High isolation between lines is inherently provided, with no direct cost (other than the fixed factor arising from the imperfect laser and detector quantum efficiencies) in terms of increased driving power.

Much of the above discussion has assumed that the electrical interconnect line is not long enough and the bandwidth not wide enough to require termination for suppression of reflections. In the event that electrical termination is required, the electrical drive power requirement increases appreciably. If the termination is perfectly matched to the characteristic impedance R_o of the line, then the power dissipation is

$$P_e = \frac{1}{2}\frac{V^2}{R_o}. \qquad (11)$$

It can now be seen that an optical interconnection will require less power than an electrical interconnection when

$$\frac{C_D + C_d}{\mathcal{R}\tau} < \frac{1}{2}\frac{V}{R_o}. \qquad (12)$$

Note that this result is independent of length for the case of lossless optical and electrical interconnects.

We defer until section 7 a more specific comparison of the power requirements for electrical and optical interconnects.

6. Fan-in and Fan-out Properties of Optical Interconnections [12]

An interconnection is said to have an *N-fold fan-out* if it provides a path from a single source of information to N different destinations. Whether the interconnections are electronic or optical, at each of the N destinations there will usually be a device capacitance that must be charged by the electrons delivered or generated by the interconnection. If N-fold fan-out is present, then the electrons or photons must be divided at least N ways (more than N ways if capacitive charging of the electronic lines or losses associated with the optical lines are considered) This fact implies that the time required to charge one device capacitance will be approximately N times as long in the presence of N-fold fan out as in the absence of fan-out. This same conclusion holds whether the interconnect paths are provided by electrons or by photons. The difference between optical and electronic interconnects with regard to fan-out resides only in the capacitances and losses associated

with the interconnect lines themselves. Low-loss optical interconnect lines do not provide the equivalent of capacitive or resistive effects associated with electronic lines.

An interconnection is said to have *N-fold fan-in* if it provides simultaneous paths from N different sources of information to a single destination. In this case, a single destination device capacitance must be charged by the N streams of electrons delivered or generated by the interconnections. If the interconnections are electronic, one source will appear to another source as a resistive path to ground, and if the resistance of such paths is sufficiently low, then a portion of the electrons delivered from any one source to the destination device may be diverted to ground through those source resistances. In fact, if the sources all have internal series resistance R_S, and if the load to which power is to be delivered has resistance R_S, then it can be shown that the fraction of power delivered by one source to the load, in the presence of N-1 other sources, is $\frac{1}{N}$ of the power delivered by that source. The fraction $\frac{N-1}{N}$ of the delivered power is dissipated in the internal resistances of the N-1 other sources.

In the optical case, a similar effect is observed as a consequence of fan-in. A basic optical fact, derivable from the laws of thermodynamics and often referred to as the *constant brightness theorem* [13], states that no passive linear optical system can increase the brightness (watts per steradian per unit area) of an optical beam. This theorem implies that any attempt to superimpose mutually incoherent beams of light in such a way that the resultant brightness would be increased must inevitably fail. More specifically, the optical system devised to accomplish this goal must have associated with it some form of loss mechanism that will prevent the brightness from being increased. A good example is afforded by a holographic optical element designed to merge two mutually incoherent

beams of light into a single beam having the same cross-sectional area and angular divergence as the original beams. Figure 5(a) shows the recording geometry that might be

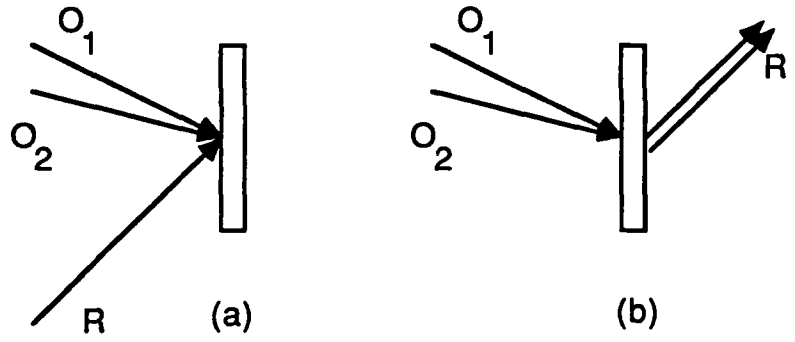


Figure 5. Holographic beam combiner. (a) Recording. (b) Utilization.

devised to make such an element. The hologram is recorded using two object beams and a single reference beam. Figure 5 (b) shows the geometry in which such an element would be used for fan-in. Two incident beams, each coinciding in direction with one of the original object beams, are merged into a single beam traveling in the direction of the original reference beam. It can be shown that such an element must have associated with it at least a 50% loss, arising from the fact that at least 50% of the light in each of the incident beams remains in the zero order of the holographic optical element. In this way the constant brightness theorem is satisfied. More generally, if an element is made to merge N different optical beams into a single beam having the same cross-sectional area and angular divergence as its component beams, then at most 1/N th of the light in each beam can be placed into the desired output beam. However, lossless fan-in can in principle be accomplished if

$$A_2\Omega_2^2 > NA_1\Omega_1^2 \qquad (12)$$

where A_1 and A_2 are the cross-sectional areas of a single input beam and the output beam, respectively, while Ω_1^2 and Ω_2^2 are the solid angles subtended by one of the N input beams and the output beam, respectively.

If the two beams to be merged are mutually coherent, then the predictions of the constant brightness theorem must be used with care. It has been shown [14] that the merging of two single-mode mutually coherent beams in a waveguide Y-junction into a single output waveguide can yield an efficiency anywhere between 0% (input waveguides driven out of phase) and 100% (input waveguides driven in phase). If the phase difference between the two coherent beams is entirely random, then each input beam is coupled into the output waveguide with an efficiency of 50%, consistent with the constant brightness theorem, the rest of the light being lost through radiation modes. Thus average power per output mode obeys the constant brightness theorem.

It should be re-emphasized that in the incoherent case the losses mentioned previously can be avoided if the resultant beam is allowed to have a sufficiently larger cross sectional area or angular divergence than the component beams. Such is usually the case if the fan-in takes place on a detector which is capable of accepting radiation from a larger solid angle than that occupied by any one of the beams. In such a case, no attempt is made to force the beams to coincide in their directions of propagation.

7. Power Comparisons for Example Electrical and Optical Interconnects

The power required to drive an interconnect line is one of several characteristics that important in deciding the superiority of one interconnect technology over another. Note that drive power alone is not a complete characterization of an interconnect problem, for it

ignores other important characteristics, such as mutual coupling between adjacent interconnect lines. To understand the power requirements is to understand but one of the dimensions in a multidimensional comparison space, albeit a very important dimension.

In this section we compare the power required for interconnects at two levels of architecture, focusing on the gate-to-gate interconnect problem, and the chip-to-chip interconnect problem. In the latter case, both unterminated and terminated electrical interconnect lines are considered.

Power Requirements in Gate-to-Gate Interconnects

We first consider the simplest of interconnection problems, illustrated in Fig. 6. Gates on a

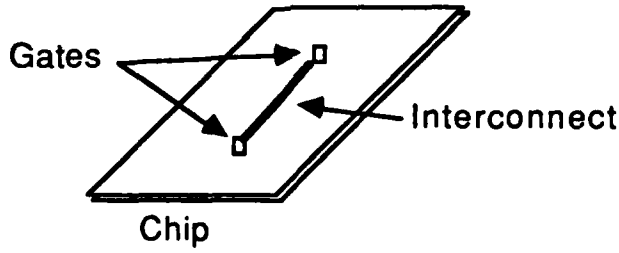


Figure 6. Gate to gate interconnect

single silicon chips are to be interconnected by a single interconnection line. In one case the interconnection will be accomplished by conventional electronic means, while in the other the interconnection will be accomplished by optical means.

In both the electronic and the optical interconnect cases, the interconnect must supply sufficient electrical charge in a clock period to charge the capacitance C_g of a destination gate to its threshold voltage V. In the process, an electrical interconnect must supply

sufficient reactive power to charge to that same voltage not only the capacitance of the destination gate, but also the capacitance C_l of the interconnect line itself, as well as the capacitance C_g of the source gate. Thus the total capacitance to be charged is

$$C_{TOT} = 2C_g + C_l. \tag{13}$$

The capacitance of a metal plate of area A separated from an infinite ground plane by a dielectric of relative dielectric constant ε_r and thickness d is

$$C = \frac{\varepsilon_r \varepsilon_o}{d} \tag{14}$$

where $\varepsilon_o = 8.854 \times 10^{-4}$ F/cm is the dielectric constant of free space. Considering first the gates, projected VLSI device lengths and oxide layer thicknesses are taken to be 0.5 and 0.02 μm, respectively. An SiO_2 oxide is assumed ($\varepsilon_r = 3.9$). The resulting gate capacitance is $C_g = 50$ fF. Considering the interconnection line itself, C_l is given by

$$C_l = \frac{\varepsilon_r \varepsilon_o w l}{h} \tag{15}$$

where l is the line length, w the line width, and h the height of the line above the ground plane. The width/height ratio is restricted by fringing field effects to a minimum value of about 2. Considering a typical length of the interconnect to be about 1 mm, the line capacitance is 70 fF. Thus the total capacitance of the gate-to-gate link is $C_{TOT} = 170$ fF. Looking to the future, we assume for the purpose of illustration that the gate capacitance must be charged in 1 nsec (implying a data rate of 1 Gb/s). Assuming a 1 volt gate threshold, Eq. (1) implies that the reactive power to charge the gate in 1 nsec is

$$P_e = 85\ \mu W \tag{16}$$

Calculation of the power required to solve this same interconnect problem using optics proceeds as follows. In this case we assume that a laser is driven by a current source, the optical power generated by that laser is coupled into a waveguide with perfect efficiency, and the optical signal is delivered without loss to a detector having a responsivity of 0.5 A/W. The detector is taken to be 2 μm thick and to be 25 μm on a side, yielding a detector capacitance of 33 fF. Thus the combined capacity C_t of the parallel detector and gate capacitances is 83 fF. The laser is assumed to have an overall power efficiency of 25%. To reach a 1 volt threshold on a capacitance of 83 fF in 1 nanosecond requires a charging current ($i = \frac{VC_t}{\tau}$) of 83 μA. The responsivity of 0.5 A/W leads to a required optical power of 166 μW at the detector. The 25% source efficiency in turn requires an electrical drive power of approximately 660 μW at the beginning of the link. Thus power required of the electro-optic link is P_{eo} = 660 mW.

The power calculated above is actually an underestimate of what would be required in practice, even for a lossless optical interconnect line. The reason lies in our neglect of the fact that, to achieve lasing action in the optical source, which is required for high overall efficiency, a certain minimum threshold current is required. As discussed previously, the smallest threshold currents for sources currently available commercially are of the order of 1 ma. Following the example presented earlier in connection with reference #9, the electrical power required to drive the optical link is modestly larger than the threshold power of 1.4 mW. Future improvements of laser devices could drive this number lower, but probably not be more than a factor of 2. Since we are looking to the future here, we assume a minimum electrical drive power of 1 mW. Hence, we use in all future comparisons a drive power required of the electro-optic link given by

$$P_{eo} = 1\text{mW}. \qquad (17)$$

We see from the above considerations that, for the parameters assumed in this example, the optical link is not competitive with the electrical link at the gate-to-gate level, assuming that drive power is the determining factor. This conclusion is likely to be true in general at this lowest level of architecture. Such is not necessarily so at the chip-to-chip level and higher, as we shall now see.

Power Requirements in Chip-to-Chip Interconnects

As illustrated in Fig. 7, in the electrical case the interconnection line must be attached to

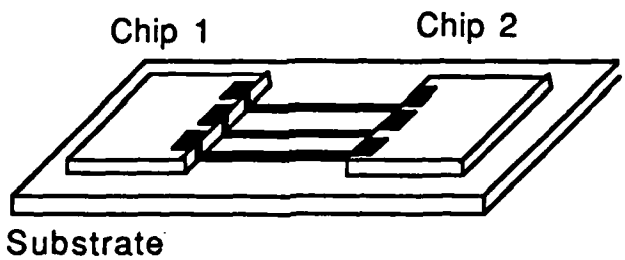


Figure 7. Chip to chip electrical interconnect

bonding pads on each of the two chips. We initially consider the case of an unterminated electrical interconnect line. To minimize propagation delays, the bonding pad is driven by a series of gates, with gate capacitances gradually increased until the device capacitance is comparable to that of a bonding pad. A current pulse from the driving logic element must be capable of charging capacitances of these larger gates, the bonding pads at both ends of the line, the line itself, and the destination gate. The total capacitance to be charged is thus

$$C_t = 2C_g + 2C_b + C_l \qquad (18)$$

where C_b is the capacitance of a bonding pad, C_g is the capacitance of a gate, and C_l is the capacitance of the metallic interconnect line. The gate capacitances have been discussed in the previous section, and again are taken to be 50 fF.

Considering the bonding pads, for a pad area of 100 μm^2, and the same SiO_2 dielectric layer assumed previously, the bonding pad capacitance is about 0.4 pF. The line connecting the bonding pads on the two chips is is assumed to be 25 μm wide, 500 μm above the ground plane, and about 1 cm in length. The resulting line capacitance is only 4.5 fF. It is assumed for the present that the line is unterminated.

Examining the capacitances discussed above, we see that the dominant capacitance is that of the bonding pads, which are by far the largest structures on the chips themselves. The total capacitance C_t is 0.9 pF, essentially equal to the combined capacitances of the two bonding pads and the two gates. Again looking to the future of silicon technology, we assume that the data link between chips will be required to operate at a rate of 1 Gb/s ($\tau = 10^{-9}$ sec). Assuming that the final gate has a threshold voltage of 1 volt, the reactive power required to drive the electrical interconnection is then given approximately by

$$P_e = 450\ \mu w. \qquad (19)$$

Turning attention to the optical solution to this interconnection problem, we replace the bonding pads and electrical line with an essentially lossless optical path (e.g. a fiber). Again the capacitance of the destination gate must be charged to the threshold voltage, taken to be 1 volt, in a time period of 1 nsec. The electro-optical parameters are all taken to

be identical to those used in the gate-to-gate illustration, yielding the same electrical power required to drive the optical interconnection,

$$P_{eo} = 1\text{mW} \qquad (20)$$

We see that in this case the amounts of power required for the electrical interconnection and the optical interconnection are not far apart. If the interconnect has some degree of fan-out associated with it, the optical link becomes even more attractive in the comparison (to double the optical power emitted by the laser requires only an increase by 1.8 in electrical drive power for this particular laser). Therefore we can conclude that in this example, the use of an optical interconnection can not be clearly rejected based on drive power considerations, but it also can not be justified on this basis. Rather, it is the immunity from mutual interference that is the main attraction of the optical link.

We turn now to the case of a terminated electrical line. The electrical interconnect line must be terminated in its characteristic impedance if the length of the line and the bandwidth of the interconnect exceed a certain limit. In simplest terms, if a reflection from the end of the line travels back to the source, and arrives there with as much as a half a bit-period of delay, then potentially the signal transmitted to the gate will have sufficient reflection noise to cause unreliable triggering of the gate. On a lossless transmission line with inductance L per unit length and capacitance C per unit length, the velocity of propagation is

$$v = \frac{1}{\sqrt{LC}} \qquad (21)$$

If the line length is l, then potential problems arise with an unterminated line when

$$l > v\tau. \qquad (22)$$

The velocity of propagation on a standard 50 ohm coax line is typically about 0.5 times the free-space velocity of light. For a 1 Gb/s data rate, the maximum allowable line length for an unterminated line would then be of the order of 15 cm.

To avoid reflections, it simply necessary to terminate the electrical line in its characteristic impedance. For the lossless LC line, that impedance is given by

$$R_o = \sqrt{\frac{L}{C}} \qquad (23)$$

The power required to drive a terminated electrical line can now be calculated with the help of Eq. (11). For a 50 ohm transmission line, a 1 volt threshold voltage, and again assuming a 1 nsec. bit period, the required power for an electrical interconnect becomes

$$P_e = 10 \text{ mW} \qquad (24)$$

Note that the presence of the terminating resistor has dramatically increased the drive power necessary for the electrical connection, to the point where the optical interconnection now has a distinct advantage in terms of required drive power.

8. Optical Clock Distribution to a VLSI Chip

As VLSI chip capabilities increase through the scaling down of feature sizes and the scaling up of chip areas, interconnection delays at the chip level are known to be rapidly becoming the dominant limitation to chip speed. Important among these chip-level interconnect problems is clock distribution, i.e. the transmission to all parts of the chip of a reference timing signal, free from differential delays that could lead to a loss of synchronism. The

only previous work on the problem of clock distribution at the chip level is that of Fried[15]. Much of what follows is based on the work of Clymer [16,17].

There are major problems associated with the distribution of signals within integrated circuit chips via conductors. These problems arise from the finite resistivity, capacitance and length of the conductors used as the signal paths, and from the limited number of layers available for routing. Chip designers can route signals over different types of conductors, each characterized by a different resistivity, and all having about the same capacitance per unit length. Aluminum conductors have the lowest resistivity, while polysilicon conductors have the highest, the two resistivities differing by two orders of magnitude. Low resistivity implies a small RC time constant for charging the line, and hence implies the smallest amount of delay per unit length. While aluminum is very desirable as an interconnect medium, many VLSI fabrication technologies support only one or two levels of metal interconnection. Aluminum is needed for the distribution of ground, supply voltages, and other special communication lines, and its use for other functions may be restricted by the need to avoid crossing conductors and the limited number of layers for wiring. Even in designs having several layers of wiring paths, the majority of chip surface area is occupied by interconnections rather than active devices.

The clock signal is used to synchronize the operations of a very large number of devices on a VLSI chip. The large number of devices that the clock distribution system must accommodate, and the wide range of distances that exist between devices create special limitations in this signal distribution problem. The finite capacitances and resistivity of the rather long wires, as well as the capacitances of the multitude of devices attached to the clock line (i.e. the very large fan-out), result in a very large loading of the clock driver. All capacitances are present in parallel and therefore add to produce a large overall capacitance

that must be driven. The large capacitive load causes a broadening of the clock pulses and slows the overall operation of the chip.

There are two design approaches that can be used to reduce the effect of capacitive loading of the clock drivers. The first is characterized by a chain of increasingly larger inverter stages. Such a strategy minimizes the overall delay through the chain of inverters. The second approach is illustrated in Fig. 8. This figure shows a hierarchical distribution

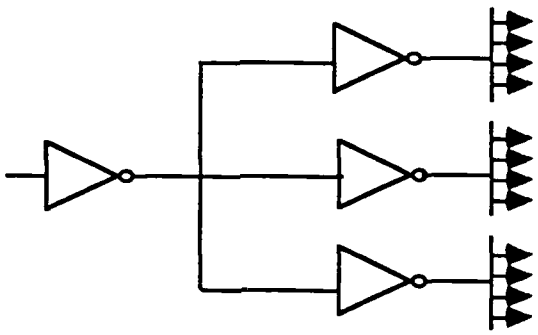


Figure 8. Hierarchical distribution with inverters.

system in which the fan-out at each node is limited to reduce the transition time for the driver stages at the nodes.

The circuit lines can be modeled as distributed RC paths, and as such the waveform propagation is governed by the diffusion equation. Each length of conductor has an associated delay that is a function of the capacitance per unit length, resistance per unit length, and length of the wire. Furthermore, as feature sizes are made smaller and smaller, the interconnection delay increases quadratically with the reciprocal of feature size for a fixed length of interconnection. The large range of different delays corresponding to the conductors connecting the clock driver to many different clocked devices leads to differences of clock pulse arrival times at those devices. Such a phenomenon is commonly

referred to as "clock skew", and it has a large influence in determining the rate at which the chip can run.

Several approaches have been suggested and implemented to reduce or eliminate the clock skew problem. One approach, suggested by Anceau [18], involves distributing a low frequency clock chipwide to several functional blocks, and internally synthesizing a high-frequency clock to synchronize operations within each block. A second approach, shown in Fig. 9, is characterized by the use of metal (heaviest lines) to distribute the clock to a multitude of smaller functional blocks, and the use of polysilicon to locally distribute the

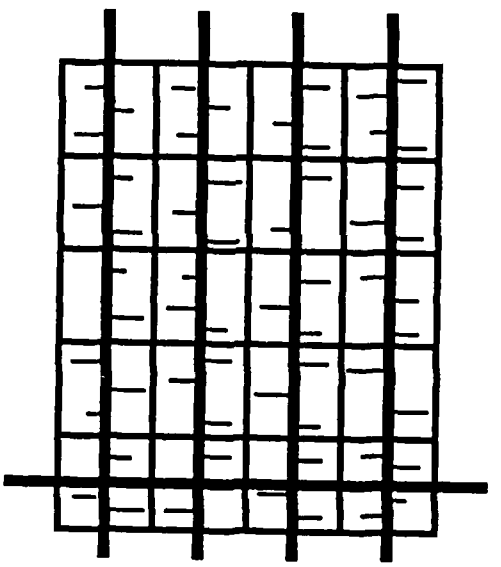


Figure 9. Hierarchical distribution system with metal and polysilicon wiring.

signal within each block. A third approach forces all lines to be of exactly the same length; one method for realizing this goal is the so-called "H-tree" distribution system shown in Fig. 10 [19]. A fourth approach eliminates a chipwide synchronization signal by designing

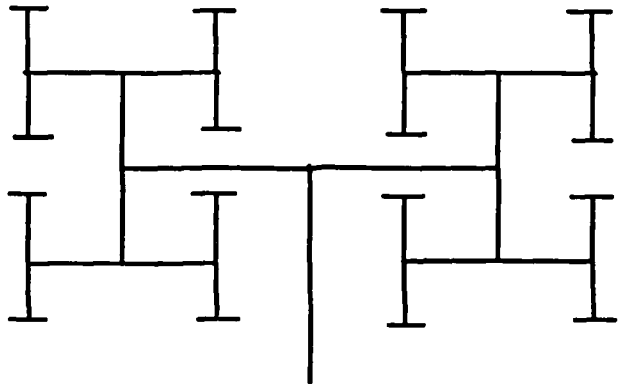


Figure 10. H-tree distribution system.

each functional block to be self-timed. This approach allows fast execution of operations within each functional block, but at the expense of handshaking delays for communication, and added control lines between functional blocks [20]. All of the above approaches have the unfortunate attribute of requiring massive use of metal wiring due to the large lengths of necessary for coverage of the entire chip.

One possible approach to the design of an optical clock distribution system is shown in Fig. 11. An optical clock signal is generated by an off-chip laser diode, shown at the top

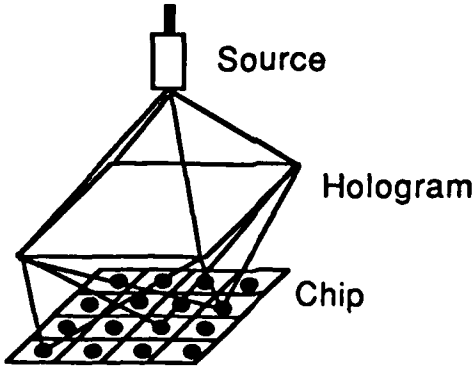


Figure 11. Optical distribution of a clock via a holographic optical element.

of the figure. The optical beam is then mapped through a holographic optical element to various photodetector sites on the chip surface. The detected signal is converted to a digital voltage on the chip, and is then distributed to nearby clocked devices. Optical clock distribution takes place to small functional cells on the chip, from which the clock signal is sent via polysilicon interconnections to the clocked devices within each functional cell. The communication delay differences from the laser source to the various detector sites are entirely negligible, and the prime source of clock skew is variations of the response times of the photodiodes and amplifiers distributed over the surface of the chip. Of course other optical methods of distributing the clock could also be envisioned, perhaps using optical waveguides on the chip itself.

Two technological problems that arise in the optical clock distribution approach should be mentioned. First, it is highly desirable to use near IR radiation for the optical clock, since the technology of high-speed semiconductor sources is well-developed in this wavelength region. However, the penetration depth of such IR radiation in silicon is greater than might be desired, leading to the generation of deep charge carriers which may be able to diffuse to nearby portions of the chip, causing unwanted interference. Design rules may have to take account of this effect, or alternatively, means for confining the charge carriers may have to be developed. Secondly, small variations in line-width across the chip, due to nonuniformities of the fabrication process, result in significant variations of time delay through the detector/amplifier circuits on the chip. A ± 1 μm variation of linewidth in the 4 μm design has been found to result in a ± 5 nsec variation of transition time of the clock waveform. This variation can be reduced if more area is devoted to the clock detector/amplifier circuitry. In addition, the assumption of a ± 1 μm variation is probably far too pessimistic.

An example of a detection and clock driver circuit is shown in Fig. 12. In this figure,

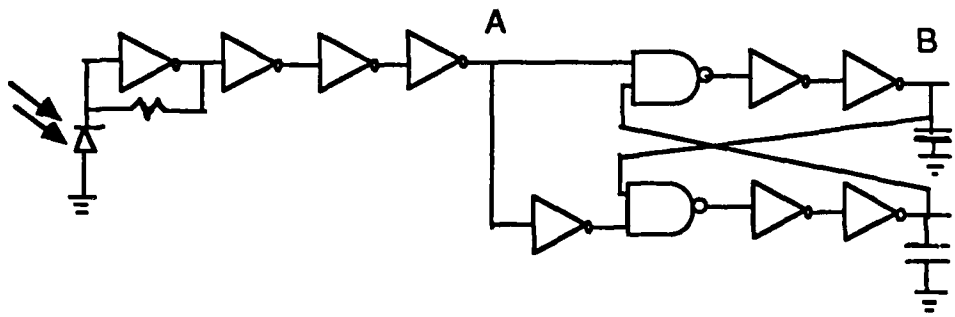


Figure 12. Detection and clock driver circuit.

devices to the left of point A represent a standard transimpedance amplifier chain commonly used in optical communication receivers [21]. The photodiode is shown on the far left. The devices to the right of point A represent a textbook example of a VLSI clock driver [22]. The outputs correspond to the two-phase clock signals commonly used in VLSI designs.

Figure 13 shows a photograph of the layout of a chip that has been fabricated for the

PHOTOGRAPH

Figure 13. Optical clock test chip.

purpose of testing the approach outlined above. The function of this chip is simply to conduct a series of tests on its own performance. The 9 vertical strips in the upper portion of the chip are the individual optical receivers, with two contained in each strip. The leftmost string of devices in the upper half of the layout comprises at test circuit for measuring the maximum allowable clock rate. The four similar vertical strips to the right of this circuit contain eight receivers used in a clock-skew measurement test. The receivers in the lower half are included to help align the optical input beams to the photodiode windows

for the leakage current tests. This chip was fabricated with the MOSIS 3 μm CMOS process. The individual detectors are 20 μm by 20 μm in size. The performance of the receivers has been tested only in the visible portion of the spectrum (632.8 nm wavelength), not yet in the near IR, where the interest is greater.

The yield obtained on this chip was very poor, due primarily to the simultaneous presence on the chip of both analog and digital circuitry. However, it was possible to find chips on which individual tests were operable, and measurements were accordingly taken. The maximum clock frequency, averaged over 10 chips, was found to be 15.1 MHz, with a standard deviation of about 1.4 MHz. A 2 μm design could be expected to have a maximum allowable clock frequency of no more than 40 MHz.

A measurement of the skew introduced by variations of the delay time experienced through different receivers was also made. The results showed an average receiver-to-receiver skew of 13.25 nsec., with a standard deviation of 1.5 nsec.

Finally, leakage current tests were performed to determine the maximum storage time achievable with dynamic latch cells that are at various distances from a photodetector. The results are a function of the optical power incident on a photodiode, but with the maximum available 632.8 nm optical power (2.2 mw) incident on a single photodetector, spacings of several tens of microns were found to be required if the storage time of the latch was to be undegraded. This constraint can be expected to be more stringent if the clocking wavelength is at 800 nm. Considerations such as these can lead to new design rules that account for the leakage currents and prevent them from degrading latch performance, at the cost of significant geometrical constraints.

The test results described above demonstrate that the transimpedance amplifier approach to clock detection and distribution suffers many limitations, and is in general not very competitive with present day non-optical approaches. For this reason, a second approach was devised that overcomes many of the above limitations. This alternative approach is now briefly described. Rather than trying to electrically amplify a a received optical clock waveform for direct use in generating an electrical clock on chip, the alternate approach uses a series of free-running digital ring oscillators wherever the transimpedance amplifiers were located in the previous approach, and uses a photodetector within a phase-locked loop circuit to force locking of each ring oscillator to the common frequency of the optically distributed clock. A circuit diagram showing the phase-locked loop is found in Fig. 14.

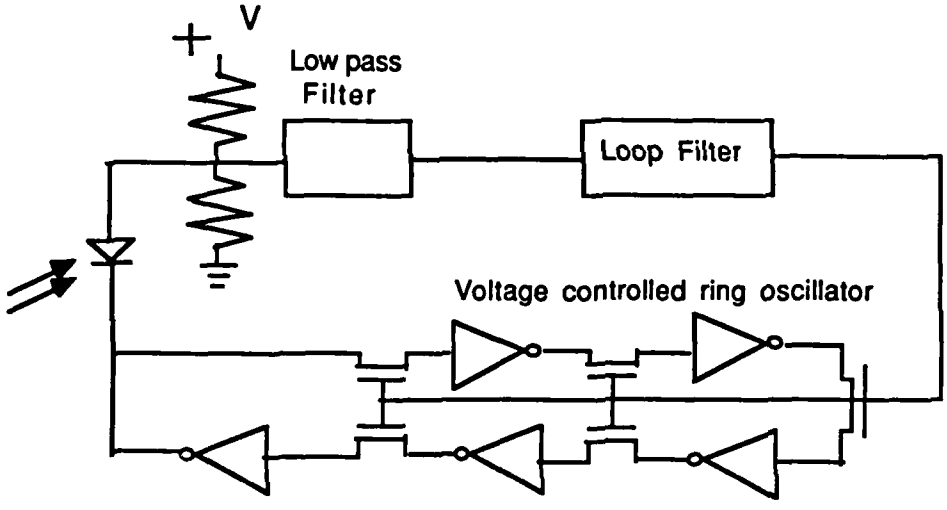


Figure 14. Phase locked loop containing an optical detector.

SPICE simulations of this approach have yielded very encouraging results, indicating that with a 2 μm CMOS process, clock frequencies up to 150 MHz should be possible. Furthermore the skew between individual frequency-locked ring oscillators is predicted to be of the order of 50 ps. Finally, far less chip area is required of a ring oscillator with a phase-lock loop than is required of a transimpedance amplifier. A CMOS chip is currently

being designed to check these predictions experimentally. Much work remains to be done on the subject of optical clock distribution at the chip level.

The understanding of the capabilities and limitations of optical clock distribution systems is still at an early stage, and it is not possible to be definitive about exactly what circumstances will justify the use of the optical approach. Undoubtedly there will be much future work in this area, not only on the problem of clock distribution at the chip level, but on the use of optics for distributing timing signals at the board and wafer levels as well.

Acknowledgements

I am indebted to several people for parts of the material presented here. Much of Section 8 is based on an analysis first performed by Raymond Kostuk. Section 9 contains work that is primarily due to Bradley Clymer. I am also grateful to Prof. E.G.S. Paige for many stimulating discussions of these and related subjects.

REFERENCES

1. Special Issue on Optical Interconnections, *Optical Engineering*, Vol. 25, No. 10, October 1986.

2. J.W. Goodman, F.J. Leonberger, S.Y. Kung, R.A. Athale, "Optical interconnections for VLSI Systems", *Proc. I.E.E.E.*, Vol. 72, No. 7, pp. 850-866, July 1984.

3. A. Husain, "Optical interconnect of digital integrated circuits and systems", *Proc. S.P.I.E.*, Vol. 466, pp. 10-20, 1984.

4. L.D. Hutcheson, P. Haugan, and A. Husein, "Optical interconnects replace hardwire", *I.E.E.E. Spectrum*, Vol. 24, No. 3, pp. 30-35, March 1987.

5. J.P. Herriau, A. Deloulbe, B. Loiseaux, J.P. Huignard, "Optical switching using photoinduced gratings", *J. Optics,* Vol. 15, No. 5, pp. 314-318, 1984.

6. J. Wilde, R. McRuer, L. Hesselink and J.W. Goodman, "Dynamic holographic interconnections using photorefractive crystals", *Proc. S.P.I.E.*, Vol. 752, pp. 200-208, 1987.

7. H. Tajima, Y. Okada, and K. Tamura, ""A high-speed optical common bus for a multiprocessor system", *Trans. Inst. Electron. and Commun. Eng. Japan.*, Vol. 24, No. 17, pp. 850-866, 1984.

8 J.M. Heaton, L. Solymar, "Reflection holograms replayed at infrared and untraviolet, *Optics Communications,* Vol. 62, N0. 3, pp. 151-154, 1987.

9 K.Y. Lau, N. Bar-Chaim, P.L. Derry, and A. Yariv, "High-speed digital modulation of ultralow threshold (< 1 mA) GaAs single quantum well lasers without bias", *Appl. Phys. Let.*, Vol. 51, No. 2, pp. 69-71 (1987).

10. R.K. Kostuk and J.W. Goodman, "Optical imaging applied to microelectronic chip-to-chip interconnects", *Applied Optics*, Vol. 24, No. 17, pp. 2851-2858, 1985.

11. Raymond K. Kostuk, *Multiple grating reflection holograms with application to optical interconnects* , Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, August 1986.

12. J.W. Goodman, "Fan-in and fan-out with optical interconnections", OPTICA ACTA, Vol. 32, No. 12, 1489-1496, 1985.

13. W.T. Welford and R. Winston, *The Optics of Nonimaging Concentrators*, Academic Press, New York, N.Y., 1978 (see Appendix A).

14. R.H. Rediker and F.J. Leonberger, *I.E.E.E. J. Quantum Electronics*, Vol. 18, pp. 1813-1816, 1982.

15. J.A. Fried, "Optical I/O for high-speed CMOS systems", *Optical Engineering*, Vol. 25, No. 10, pp. 1132-1141, 1986.

16. B.D. Clymer and J.W. Goodman, "Optical clock distribution to silicon chips", *Optical Engineering*, Vol. 25, No. 10, pp.1103-1108, 1986.

17. Bradley D. Clymer, *Optical Clock Distribution for VLSI*, Doctoral Dissertation, Department of Electrical Engineering, Stanford University, August 1987.

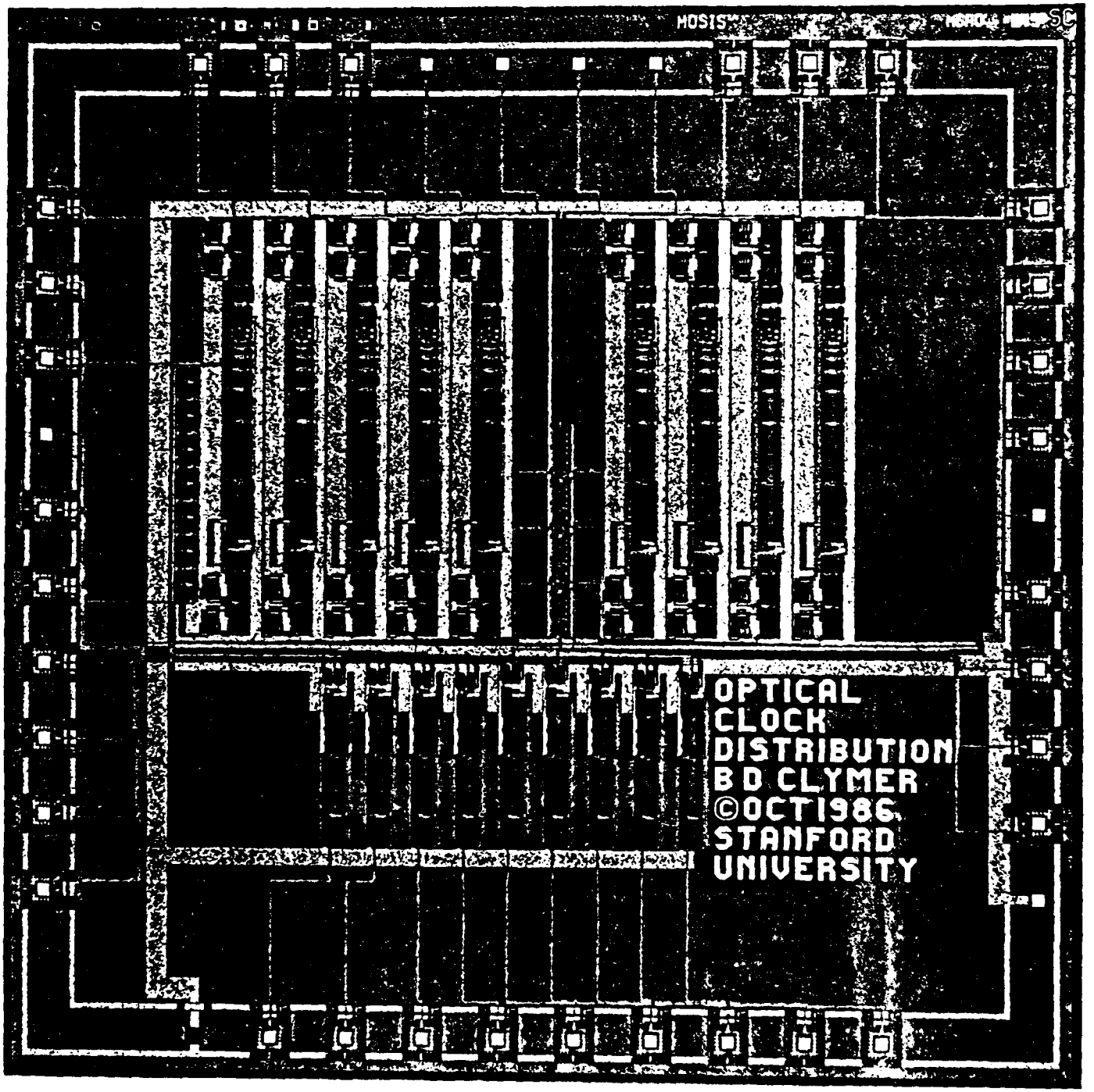
18. F. Anceau, "A synchronous approach for clocking VLSI systems", *I.E.E.E. J. Solid State Circuits*, Vol. SC-17, No. 1, pp. 51-56, 1982.

19. S. Dhar, M.A. Franklin, and D.F. Wann, "Reduction of clock delays in VLSI structures", *Proc. IEEE 1984 Conf. on Computer Design: VLSI in Computers*, pp. 778-783, 1984.

20. Carver Mead and Lynn Conway, *Introduction to VLSI Systems*, Addison-Wesley, 1980, pp. 242-246.

21. A.A. Abidi, "Gigahertz transresistance amplifiers in fine line NMOS", *J. Solid-State Circuits*, Vol. SC-19, No. 6, pp. 986-994, 1984.

22. Carver Mead and Lynn Conway, *op. cit.*, Chapter 7.


MOSIS
OPTICAL
CLOCK
DISTRIBUTION
B D CLYMER
©OCT1986,
STANFORD
UNIVERSITY

B

Neural networks for computation: number representations and programming complexity

Mitsuo Takeda and Joseph W. Goodman

a reprint from Applied Optics
volume 25 number 18, September 15, 1986

Neural networks for computation: number representations and programming complexity

Mitsuo Takeda and Joseph W. Goodman

Methods for using neural networks for computation are considered. The success of such networks in finding good solutions to complex problems is found to be dependent on the number representation schemes used. Redundant schemes are found to offer advantages in terms of convergence. Neural networks are applied to the combinatorial optimization problem known as the Hitchcock problem and signal processing problems, such as matrix inversion and Fourier transformation. The concept of programming complexity is introduced. It is shown that for some computational problems, the programming complexity may be so great as to limit the utility of neural networks, while for others the investment of computation in programming the network is justified. Simulations of neural networks using a digital computer are presented.

I. Introduction

Even the fastest modern computer cannot compare to the brain of an infant in the performance of intelligent information processing such as image processing and pattern recognition. This well-quoted fact suggests the possibility of a quite different type of computer. The fundamental difficulty in creating artificial intelligence on conventional digital computers comes from the large difference in architectures of information processing between digital computers and human brains, i.e., the sequential processing in von Neumann machines and the massively parallel computation in human brains.[1] Neuroscientists have revealed that the massive parallelism and the computational richness in the human brain lie in the global and dense interconnections among a large number of identical logic elements or neurons which are connected to each other with variable strengths by a network of synapses.[2] An artificial neural network system that can perform parallel computation and the function of natural intelligence is extremely attractive as a future-generation computer.

However, there exist two major problems that must be attacked before the realization of such a neural computer. The first is a hardware problem of how to implement those global and dense interconnections among many neuronlike logic elements, and the second is a software problem of how to program such highly parallel computation on a neural network system. We may take two different approaches to the first problem, VLSI-based interconnections and optical interconnections.[3] Neurons in the human brain are interconnected in 3-D space since it is the most natural and efficient way of interconnection, but VLSI-based interconnections are inherently 2-D in nature. Optical signals, on the other hand, can flow through 3-D space to achieve the required interconnects between neuronlike logic elements. Based on this idea, several schemes of optical computing have been proposed.[4-7] Among them, Psaltis and Farhat[4,7] recently reported an optical implementation of the Hopfield neural network[8,9] using an optical vector–matrix multiplier[10] as a programmable interconnector and demonstrated the feasibility of optical content addressable associative memory.

Extensive studies have been done on the basic characteristics of the neural networks themselves,[11] but the second problem of how to program them to do various computations of practical interest has not been fully studied except in their application to associative memory.[12] Quite recently, Hopfield and Tank[13] showed that a certain class of optimization problem can be programmed and solved on their neural network model. They demonstrated the computational power and speed of their neural network by solving one of the NP-complete problems[14] known as the Traveling-Salesman problem. The purpose of this paper is to extend their idea and explore new possibilities of programming and solving on neural networks other various nonbiological problems of practical interest. We emphasize that our goal is not to propose mechanisms that might actually be utilized by the brain but rather

The authors are with Stanford University, Department of Electrical Engineering, Stanford, California 94305.

Received 22 March 1986.
0003-6935/86/183033-14$02.00/0.

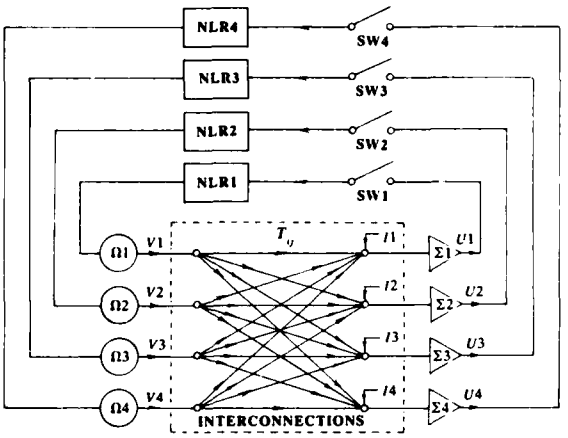


Fig. 1. Neural network model.

to apply neural network ideas to computational problems and thereby to open some new avenues for realizing powerful man-made computers.

We first review briefly the Hopfield neural network model and describe some minor modifications. Next, we propose a new scheme to represent numbers by neuron state variables, which is essential in solving numerical problems on neural networks. Based on this number representation scheme, we show how we can program and solve combinatorial optimization problems[15] known as network flow problems[16] or more specifically as the Hitchcock problem[17] and simulate its computational performance on a digital computer. Then we give a programming scheme to perform signal processing for signal recovery, such as the computations of matrix inversion and Fourier transformation. The performance is again simulated on a digital computer.

The important idea of programming complexity is then introduced, and it is shown that for some problems the data-dependent programming complexity is so great that computations invested in finding the right neural interconnection and bias patterns may equal the complexity involved in solving the problem directly without a neural network. For such problems, neural networks, as we now understand them, may not be an appropriate architecture for computational problem solving.

We conclude with the discussion of the limitations and problems that remain to be solved.

II. Hopfield Model and Its Modifications

A. Hopfield Model

The Hopfield model[8,9] consists of a number of mutually interconnected nonlinear devices called neurons whose states are characterized by their outputs V_i (which may take values between 0 and 1). The dynamics of neurons in the Hopfield model can be described in both discrete and continuous spaces.

The discrete model is illustrated in Fig. 1. At fan-in terminals Σ_i, each neuron i receives inputs $T_{ij}V_j$ from other neurons j and a bias input I_i associated with itself:

$$U_i = \sum_{j=1}^{N} T_{ij}V_j + I_i, \tag{1}$$

where N is the number of neurons, and T_{ij} are elements of an interconnection matrix representing the strengths of connections. At discrete times, switches SW_i turn on, and the inputs U_i are fed back to corresponding neurons to change their states or to leave their states fixed according to a threshold rule determined by nonlinear operators NLR_i so that

$$V_i(k+1) = \mathrm{stp}[U_i(k)], \tag{2}$$

where k is discrete time, and $\mathrm{stp}(x)$ is a unit step function which is 1 for $x \geq 0$ and 0 for $x < 0$. Thus neurons take binary values either 1 or 0, and the binary outputs are sent out from fan-out terminals Ω_i and distributed through the interconnection network to regenerate new inputs at the fan-in terminals Σ_i.

In the continuous model, neurons change their states according to the following equations of dynamics:

$$dU_i/dt = \sum_{j=1}^{N} T_{ij}V_j + I_i, \tag{3}$$

$$V_i = g(U_i), \tag{4}$$

where t is continuous time, and $g(x)$ is a nonlinear function whose form can be taken to be

$$g(x) = (1/2)[1 + \tanh(x/x_0)], \tag{5}$$

which approaches a unit step function as x_0 tends to zero.

Hopfield[9] has shown that if $T_{ij} = T_{ji}$, neurons in the continuous model always change their states in such a manner that they minimize an energy function defined by

$$E \equiv -(1/2)\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i \tag{6}$$

and stop at minima of this function. The same is also true[8] for neurons in the discrete model if we further assume that $T_{ii} = 0$.

B. Neuron Transition Modes

We adopt the discrete-time model because it is much easier to simulate on a digital computer. But when $T_{ij} \neq 0$, the model sometimes shows an oscillatory behavior or keeps wandering around the state space near the minima of the energy function. Most problems of practical interest require self-feedbacks ($T_{ii} \neq 0$) when programmed on a neural network. We, therefore, need to design transition modes that reduce such phenomena. Without claiming any similarity to natural neuron transition rules, we choose four different discrete-time transition modes for examination.

1. *Direct Synchronous Transition Mode*

All the transitions occur simultaneously when the switches SW_i turn on in synchronism at discrete times k. The fan-in inputs are directly fed back to generate new neuron states. A continuous nonlinear function

$g(x)$ allows neurons to take state values between 0 and 1. The following equations are assumed to hold:

$$U_i(k) = \sum_{j=1}^{N} T_{ij} V_j(k) + I_i; \tag{7}$$

$$V_i(k+1) = g[U_i(k)]. \tag{8}$$

2. Differential Synchronous Transition Mode

The differential equations in the continuous model are approximated by difference equations. Transitions occur synchronously. In this case,

$$\begin{aligned} U_i(k) - U_i(k-1) &= \sum_{j=1}^{N} T_{ij} V_j(k) + I_i; \\ V_i(k+1) &= g[U_i(k)]. \end{aligned} \tag{9}$$

This mode requires one memory cell for each neuron to keep its previous input.

3. Direct Asynchronous Transition Mode (Random Delays)

This mode is similar to mode 1, but the switches SW_i turn on and off asynchronously, i.e., with random delays. In this case,

$$\begin{aligned} U_i(k - \Delta t_i) &= \sum_{j=1}^{N} T_{ij} V_j(k - \Delta t_i) + I_i, \\ V_i(k - \Delta t_i + \epsilon) &= g[U_i(k - \Delta t_i)], \end{aligned} \tag{10}$$

where Δt_i are skews caused by time delays in the network and are fractions of one clock time, while ϵ is a small positive constant. Without loss of generality we can assume

$$\Delta t_1 \le \Delta t_2 \le \ldots \le \Delta t_N,$$

because the numbering of neurons is arbitrary. In this mode, one particular neuron i need not wait for the last neuron N for synchronization, and when it decides its new state, it can make use of information about new states of other neurons that have already renewed their states.

4. Differential Asynchronous Transition Mode (Random Delays)

This is an asynchronous version of mode 2. In this case,

$$\begin{aligned} U_i(k - \Delta t_i) - U_i(k - \Delta t_i - 1) &= \sum_{j=1}^{N} T_{ij} V_j(k - \Delta t_i) + I_i, \\ V_i(k - \Delta t_i + \epsilon) &= g[U_i(k - \Delta t_i) - U_i(k - \Delta t_i - 1)]. \end{aligned} \tag{11}$$

Using simulations on a digital computer, we found that the synchronous transition modes (1) and (2) gave rise to large oscillations in the energy function when $T_{ii} \neq 0$ but that the asynchronous transition modes (3) and (4) have greatly reduced oscillatory or wandering behavior, although the reduction is not complete. While mode (3) is quicker in minimizing the energy function, mode (4) has more reduced oscillations. Depending on the characteristics of the problems of interest, we shall make a proper choice of a mode from (3) and (4).

III. Number Representation Schemes

In most problems of practical interest, solutions are described by a set of numbers. Therefore, we must have a means to encode numbers on neuron state variables V_i. While allowing neurons to take continuous state values during the process of energy function minimization, we demand that they take binary values of 1 or 0 at the final stage so that we can obtain digital solutions like those given by digital computers. For simplicity, we first assume the numbers are positive integers including 0, although we can also represent general bipolar and complex numbers by using additional neurons. We consider three different ways of mapping the positive integer space Z^+ onto the neuron state space V.

A. Binary Scheme

A common way of representing numbers in digital computers is to use binary digits. For example, 5 is expressed by 0101. This scheme uses $\log_2(N+1)$ bits to express a number N. If we let one neuron represent 1 bit, we have a one-to-one correspondence between elements in the number space Z^+ and those in the neuron state space V. Despite the economy in the number of bits or neurons used, a system based on the binary scheme is not fault-tolerant. In other words, even a single failure in a highly significant bit gives rise to a large error in the number represented.

B. Simple Sum Scheme

In this scheme, a number is represented by a simple sum of the neuron state variables V_i, i.e., the total number of firing ($V_i = 1$) neurons. For example, 5 is expressed by 0011111, 0101111, 1101011, all of which have five 1-bits. This is a one-to-many mapping from Z^+ to V, and the numbers have degenerate representations. This scheme requires N bits to express a number N and is not economical in the number of bits or neurons. However, it is highly fault-tolerant because an error in a single bit does not cause a large error in the number represented. The fault-tolerance of the human brain is believed to come from this type of averaging over a large number of neurons.[11]

So far, we have compared the binary scheme and the simple-sum scheme from the viewpoint of their fault-tolerance. More important is their difference in problem-solving capability. As will be seen, problems are solved through a spontaneous energy minimization process in a neural network, and the solution is given by a point in the neuron state-variable space that is reached after this minimization process. In the binary scheme, there is only one point in the state variable space that gives a correct solution. In the simple-sum scheme, on the other hand, multiple points give the correct solution. Because of this degeneracy and the clustering of quasi-minimum energy points in the neuron state-variable space, the simple-sum scheme offers more chances to reach the correct solution. Suppose,

for example, 3 is the correct solution. In the simple-sum scheme, we can get a correct solution when the final state is either 00111, 10110, 11100, or 10101 etc., whereas we can get the correct solution in the binary scheme only when the final state is 00011. Simulation results reported later in this paper support the hypothesized superiority of the simple-sum scheme.

C. Group-and-Weight Scheme

Despite its merit in fault-tolerance and computational capability, the simple-sum scheme requires too many neurons when solutions include large numbers. We propose the group-and-weight scheme which lies between the binary and the simple-sum schemes. In this scheme, we divide the total q bits into K groups, each of which has M bits ($q = KM$) and interpret the groups as digits whose numbers are given by simple sums of the bits in the corresponding groups. For example, with $q = 6, K = 2, M = 3$, 5 is expressed either by 100 100 $[4^1 \times (1 + 0 + 0) + 4^0 \times (1 + 0 + 0) = 5]$, 010 001, 001 010, or 100 001 etc. A number expression for the simple-sum scheme is given by

$$\sum_{k=1}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M}V_{(k-1)M+i}\right]. \tag{12}$$

The expression includes the binary and the simple-sum schemes as special cases. When we put $M = 1$ and $K = q$, we obtain a number expression for the binary scheme

$$\sum_{k=1}^{q}2^{k-1}V_k, \tag{13}$$

and when we put $M = q$ and $K = 1$, we obtain a number expression for the simple-sum scheme

$$\sum_{i=1}^{q}V_i. \tag{14}$$

The group-and-weight scheme requires $M \log_{M+1}(N + 1)$ bits to express a number N. This also gives the number of bits required in the binary scheme when we put $M = 1$ and that required in the simple-sum scheme when we put $M = N$.

D. Bipolar and Complex Integers

So far, we have restricted our number representations to positive integers, but they can easily be extended to include bipolar and complex integers. A bipolar expression can be obtained simply by adding a negative bias integer to the expression for positive integers given by Eq. (12):

$$\sum_{k=1}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M}V_{(k-1)M+i}\right]-\lfloor(1/2)[(M+1)^K-1]\rfloor, \tag{15}$$

where $(1/2)[(M + 1)^K - 1]$ is half of the largest positive integer that can be expressed by Eq. (12), and the floor operation $\lfloor x \rfloor$ gives the nearest integer value less than x. Equation (15) can express bipolar integers ranging over $\pm\lfloor(1/2)[(M + 1)^K - 1]\rfloor$.

To express complex integers, we need twice as many neurons, i.e., neurons $V_i^{(R)}$ and $V_i^{(I)}$ that represent real and imaginary parts, respectively. Complex integers are expressed by

$$\sum_{k=1}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M}V^{(R)}_{(k-1)M+i}\right]-\lfloor(1/2)[(M+1)^K-1]\rfloor + j\left\{\sum_{k=1}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M}V^{(I)}_{(k-1)M+i}\right]-\lfloor(1/2)[(M+1)^K-1]\rfloor\right\}, \tag{16}$$

where $j^2 = -1$.

E. General Real and Complex Numbers

We can also express numbers with fractional digits, e.g., 13.26, 3.14, by using more neurons and labeling them with negative subscripts ($i < 0$), e.g., V_{-4}, V_{-12}, etc., so that the parameter k in the first summation in Eq. (12) can run from a negative integer $-K'$; the number representation becomes

$$\sum_{k=-K'}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M}V_{(k-1)M+i}\right]. \tag{17}$$

Equation (17) can express numbers ranging from 0 to $(M + 1)^K - (M + 1)^{-(K'+1)}$, with a minimum digit of quantization being $(M + 1)^{-(K'+1)}$. Just as we did in Sec. III.D, we can easily modify Eq. (17) to a form similar to Eq. (16), so that it can express general complex numbers. Again here, the group-and-weight scheme includes the binary and simple-sum schemes as special cases. If we put $M = 1$ and $K = q$, Eqs. (15), (16), and (17) give the expressions for the binary scheme. Likewise, the expressions for the simple-sum scheme can be obtained by substituting $M = q$ and $K = 1$ into Eqs. (15) and (16) and $M = q$ and $K = -K'$ into Eq. (17).

Finally, it should be noted that the number representation schemes we proposed here are all based on linear mapping of the number space onto the neuron state space. In other words, numbers are represented by linear combinations of neuron state variables. This is an important point in designing number representation schemes for the Hopfield neural network, since the energy function Eq. (6) has a quadratic form with respect to neuron state variables. Other nonlinear mapping schemes, like floating-point expressions, cannot form the energy function required by the Hopfield model, because the floating-point expressions need to have neuron state variables in exponents. This certainly limits the possibility of covering a wide range of numbers using a small number of neurons, but for a neural computer it is not a fatal disadvantage because the use of ample neurons with much redundancy is the key to improving its computational capability and system stability.

IV. Hitchcock Problem

Based on the number representation schemes described in the previous section, we show how a combinatorial optimization problem known as the Hitchcock problem[17] can be programmed and solved on a neural network.

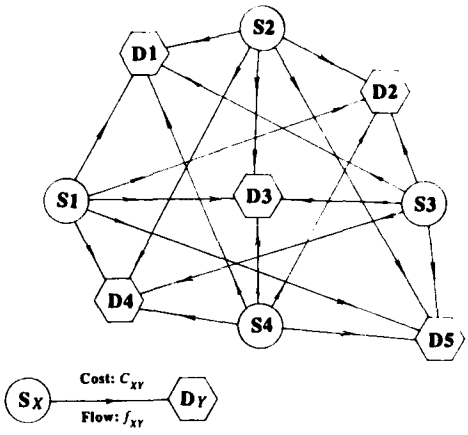


Fig. 2. Hitchcock problem with four sources and five demands.

Suppose there are m sources ($X = 1, \ldots, X = m$) for a commodity, with S_X units of supply at X, and n sinks ($Y = 1, \ldots, Y = n$) for the commodity, with a demand D_Y at Y, as shown in Fig. 2. If C_{XY} is the unit cost of shipment from X to Y, the Hitchcock problem is to find a flow f_{XY} that satisfies demands for supplies and simultaneously minimizes flow cost. Thus the problem is to minimize

$$\sum_{X=1}^{m}\sum_{Y=1}^{n} C_{XY} f_{XY}, \tag{18}$$

under the constraints

$$\sum_{Y=1}^{n} f_{XY} = S_X \qquad (X = 1,2, \ldots, m), \tag{19}$$

$$\sum_{X=1}^{m} f_{XY} = D_Y \qquad (Y = 1,2, \ldots, n). \tag{20}$$

In Table I, (a) is an example of a unit cost table, and (b) is an example of a solution represented in the form of a flow matrix or a transportation matrix. The flow matrix describes, for example, that from the source at $X = 2$, two units of the commodity should be sent to the demand at $Y = 1$ and one unit to the demand at $Y = 2$.

A. Flow Matrix Representation

Table II shows how the flow matrix can be represented by neurons. We assign q neurons to each matrix element to represent its contents f_{XY}, so that we use $N = qmn$ neurons in total for the complete representation of the flow matrix. For the convenience of mathematical treatment, we specify each neuron by a set of three subscripts $V_{XY,i}$, where XY specifies the matrix element the neuron belongs to, and i specifies the position of the neuron in that matrix element. Since the group-and-weight number representation scheme includes the binary and simple sum schemes as special cases, we express the flow matrix elements f_{XY} by the group-and-weight scheme:

$$f_{XY} = \sum_{k=1}^{K}\left[(M+1)^{k-1}\sum_{i=1}^{M} V_{XY,(k-1)M+i}\right]. \tag{21}$$

Table I. (a) Cost Matrix for the Hitchcock Problem; (b) Sample Solution Depicting the flow from Source X to Demand Y

Cost Matrix, $C_{X,Y}$					
	$Y=1$	$Y=2$	$Y=3$	$Y=4$	$Y=5$
$X=1$	5	1	7	3	3
$X=2$	2	3	6	9	5
$X=3$	6	4	8	1	4
$X=4$	3	2	2	2	4

(a)

Flow Matrix, $f_{X,Y}$						
		D1	D2	D3	D4	D5
		2	7	3	2	4
S1	5	0	5	0	0	0
S2	3	2	1	0	0	0
S3	4	0	0	0	2	2
S4	6	0	1	3	0	2

(b)

Table II. Neural Representation of the Flow Matrix for the Hitchcock Network Flow Problem; q Neurons are used to Represent One Element of the Flow Matrix

The Hitchcock Problem															
	$Y=1$				$Y=2$				⋯⋯⋯				$Y=n$			
	i				i				⋯⋯⋯				i			
	q	⋯	2	1	q	⋯	2	1					q	⋯	2	1
$X=1$																
$X=2$																
⋮																
$X=m$																

$V_{2,1,2}$ $\qquad$ $f_{1,2}$

B. Energy Function

We use the spontaneous energy minimization process of a neuron network to solve optimization problems. Since the energy function defined by Eq. (6) has a quadratic form with respect to neuron state variables V_i, we find a quadratic function of $V_{XY,i}$ so that the minimization of the function corresponds to minimizing the flow cost and minimizing violations of the constraints. An energy function that satisfies such requirements is given by

$$\begin{aligned} E = {} & -(A/2)\sum_{X=1}^{m}\sum_{Y=1}^{n}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}\left[1 - 2V_{XY,(k-1)M+i}\right]^2 \\ & + (B/2)\sum_{X=1}^{m}\left[S_X - \sum_{Y=1}^{n}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}V_{XY,(k-1)M+i}\right]^2 \\ & + (C/2)\sum_{Y=1}^{n}\left[D_Y - \sum_{X=1}^{m}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}V_{XY,(k-1)M+i}\right]^2 \\ & + (D/2)\left[\sum_{X=1}^{m}\sum_{Y=1}^{n}\sum_{k=1}^{K}\sum_{i=1}^{M}C_{XY}(M+1)^{k-1}V_{XY,(k-1)M+i}\right]^2, \end{aligned} \tag{22}$$

where A, B, C, and D are positive weight factors. The

first term weighted by A is introduced for the binarization of the neuron state variables $V_{XY,i}$, i.e., $V_{XY,i} = 1$ or 0. Because the function $F(V) = -(1 - 2V)^2$, $(0 \le V \le 1)$ takes minimum values at $V = 0$ and $V = 1$, minimizing this term assures that the final solution is given by binary numbers. The second term, weighted by B, is introduced to minimize violations of the source constraints given by Eq. (19). Likewise, through minimization of the third term with a weight C, we can satisfy the demand constraints given by Eq. (20). The last term, weighted by D, is for minimization of the total flow cost. The total cost is squared in Eq. (22), but we may also introduce it without squaring, because the cost is always positive. Note that the way we define the energy function is not unique, so that we can solve the same problem by using different programs on the neural network, just as is often the case in solving problems on conventional digital computers.

Considering the various terms represented in Eq. (22), it can be seen that solutions with low energy do not necessarily correspond to solutions with low cost. However, if the weighting constants are properly chosen, the binarization, source, and demand constraints will eventually all be perfectly satisfied, resulting in a one-to-one relation between energy and cost. Thus eventually low-energy solutions will correspond to low-cost solutions.

C. Interconnection Matrix

By analogy with digital computers, if we regard the expression for the energy function Eq. (22) as a source program, the next step is to compile or map it onto the interconnection strengths T_{ij} of the neural network. This can be done by comparing Eq. (22) with the energy function Eq. (6), which is now written as

$$E = -(1/2) \sum_{X=1}^{m} \sum_{Y=1}^{n} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{X'=1}^{m} \sum_{Y'=1}^{n} \sum_{k'=1}^{K} \sum_{i'=1}^{M} T_{XY,(k-1)M+i;X'Y',(k'-1)M+i'} \times V_{XY,(k-1)M+i} V_{X'Y',(k'-1)M+i'} - \sum_{X=1}^{m} \sum_{Y=1}^{n} \sum_{k=1}^{K} \sum_{i=1}^{M} V_{XY,(k-1)M+i} I_{XY,(k-1)M+i}, \quad (23)$$

where $T_{XY,(k-1)M+i;X'Y',(k'-1)M+i'}$ denotes the strength of the interconnection between the neuron at the $[(k - 1)M + i]$th position in the flow matrix element at XY and the neuron at the $[(k' - 1)M + i']$th position in the flow matrix element at $X'Y'$. By equating the corresponding coefficients of the two quadratic equations (22) and (23), we can determine the interconnection strengths and the biases:

$$T_{XY,(k-1)M+i;X'Y',(k'-1)M+i'} = 4A(M + 1)^{k-1}\delta_{XX'}\delta_{YY'}\delta_{kk'}\delta_{ii'} - B(M + 1)^{k+k'-2}\delta_{XX'} - C(M + 1)^{k+k'-2}\delta_{YY'} - D(M + 1)^{k+k'-2}C_{XY}C_{X'Y'}, \quad (24)$$

$$I_{XY,(k-1)M+i} = -2A(M + 1)^{k-1} + B(M + 1)^{k-1}S_X + C(M + 1)^{k-1}D_Y, \quad (25)$$

where $\delta_{ZZ'}$ is a Kroneker delta defined by

$$\delta_{ZZ'} = \begin{cases} 1 & (Z = Z') \\ 0 & (Z \neq Z') \end{cases}$$

In Eq. (24), the first term describes self-feedbacks, the second and third terms represent local interconnections between neurons in the same row ($X' = X$) and in the same column ($Y' = Y$), respectively. The last term describes the global interconnections between all neurons. If we put $M = 1$ and $K = q$, we obtain the interconnection strengths and the biases for the binary number representation scheme:

$$T_{XY,k;X'Y',k'} = 4A2^{k-1}\delta_{XX'}\delta_{YY'}\delta_{kk'} - B2^{k+k'-2}\delta_{XX'} - C2^{k+k'-2}\delta_{YY'} - D2^{k+k'-2}C_{XY}C_{X'Y'}, \quad (26)$$

$$I_{XY,k} = -A2^k + B2^{k-1}S_X + C2^{k-1}D_Y. \quad (27)$$

Likewise, the interconnection strengths and the biases for the simple-sum scheme can be obtained by putting $M = q$ and $K = 1$:

$$T_{XY,i;X'Y',i'} = 4A\delta_{XX'}\delta_{YY'}\delta_{ii'} - B\delta_{XX'} - C\delta_{YY'} - DC_{XY}C_{X'Y'}, \quad (28)$$

$$I_{XY,i} = -2A + BS_X + CD_Y \quad (29)$$

D. Numerical Experiments

To examine the computational performance of a neural network, we simulated state transitions of neurons by using a digital computer. We used the unit costs and the source and demand constraints listed in Table I. Based on these data, we determined the interconnection strengths and biases. Since at present we have no systematic methods for finding the best combination of the weighting factors A, B, C, and D, they were found empirically through the observation of several experimental results. The lack of a systematic method for finding the weighting factors should not be too disturbing. Such a situation is commonly encountered in solving multiple-target optimization problems (on a conventional digital computer), such as lens design problems and color matching problems. However, it should be emphasized that the ability to obtain a good solution depends strongly on making good choices for A, B, C, and D. Throughout the experiments with the Hitchcock problem, we used the direct asynchronous transition mode and the nonlinear function given by Eq. (5) with $0.1 \le x_0 \le 1$.

Figure 3 shows an example of the reduction of energy performed by a network with $N = 60$ neurons that represent the flow matrix based on the binary number representation scheme ($N = qmn = 3 \times 4 \times 5 = 60$, $M = 1$, $K = 3$). Table III shows the flow matrices obtained at several points on the curve of Fig. 3. The weight factors were chosen as $A = 27$, $B = C = 80$, and $D = 0.2$. Since we have no *a priori* knowledge about the solution, uniformly distributed random numbers between 0 and 1 were generated and assigned to the initial states of the neurons. Starting from a very high energy state, the neural network reduced its energy spontaneously by changing its state so that the flow matrix could satisfy the constraints while minimizing the total cost. After six iterations, we reached feasible solu-

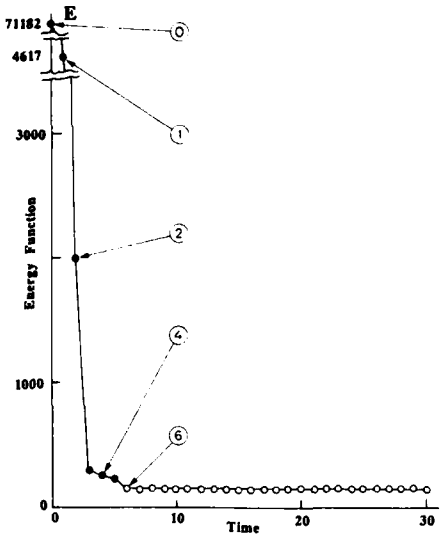


Fig. 3. Neural dynamics for the Hitchcock problem using a binary number representation scheme. The initial states are randomly generated from a seed, and the transition mode is direct asynchronous. The final transportation matrix gives a network flow cost of 40. The constraints in the energy function are chosen as $A = 27, B = C = 80$, and $D = 0.2$. The constant x_0 is 0.5. See Table III for the flow matrices at the iteration numbers indicated by the arrows.

tions (marked by open circles) that satisfied all the constraints and gave 40 as the total cost.

After arriving at a solution using the neural network, it is important to develop some understanding of how good that solution might be. To achieve this end, one could enumerate all the feasible solutions that satisfy the constraints and from this set determine the best solution. However, since it is very hard to enumerate all the solutions of underdetermined simultaneous integer equations, Eqs. (19) and (20) (which belong to a family of Diophantine equations), we used a Monte Carlo method and found 50,000 feasible solutions. (Note that this calculation was performed simply to check how well the neural network had performed.) Figure 4 shows a cost histogram of the feasible solutions found. The solution with cost 40 is found to be one of the very good solutions, which would be reached only with a probability of 6×10^{-5} if we searched randomly among the feasible solutions. Yet it is still not the best solution, which was confirmed to be 38 by using a stepping stone algorithm.

Figure 5 and Table IV show another example, for which we assigned 0.5 to the initial states of all neurons so that they started evolving from the fuzziest states. In this example, we reached a feasible solution with cost 49 at the seventh iteration, but we could not reach any other feasible solutions by further iterations. The oscillatory behavior of the energy function arises from using a discrete model with self-feedback. The solution with cost 49 is fairly good but not as good as in the previous example. Experiments performed with different initial values and/or weight factors gave solutions most frequently with costs around 50 and could not pick up the best solution. In worst cases, no feasible solution could be reached. These results are indicative of the limitations of the problem-solving capability of the binary number representation scheme. As we now show, much better results can be obtained with a degenerate number representation scheme.

Table III. Flow Matrices for the Specified Numbers of Iterations Corresponding to the Points Indicated on Fig. 3

No.0	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	3.6	5.2	6.3	4.6	2.8
S2 3.0	1.2	3.4	1.4	5.3	2.4
S3 4.0	2.9	3.4	5.8	4.0	2.5
S4 6.0	5.5	1.6	5.9	3.0	1.3

(a)

No. 1	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	0	0	0	0
S2 3.0	0	0	0	0	0
S3 0	0	0	0	0	5.0
S4 6.0	0	1.0	3.0	0	2.0

(b)

No. 2	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	3.0	7.0	0	0	0
S2 3.0	1.0	1.0	1.0	0	0
S3 4.0	0	0	0	0	4.0
S4 6.0	1.0	1.0	3.0	2.0	0

(c)

No. 4	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	1.0	5.0	0	0	0
S2 3.0	1.0	1.0	0	0	0
S3 4.0	0	0	0	0	4.0
S4 6.0	0	1.0	3.0	2.0	0

(d)

No. 6	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	5.0	0	0	0
S2 3.0	2.0	1.0	0	0	0
S3 4.0	0	0	0	0	4.0
S4 6.0	0	1.0	3.0	2.0	0

(e)

To examine the problem-solving capability of the degenerate number representation schemes, we programmed the same problem on a 140-neuron network using the simple-sum scheme ($N = qmn = 7 \times 4 \times 5 = 140, M = 7, K = 1$). Figures 6 and 7 and Tables V and VI show the computational performance of the 140-neuron network with its initial states all set equal to 0.5, the fuzziest states. Weight factors were chosen to

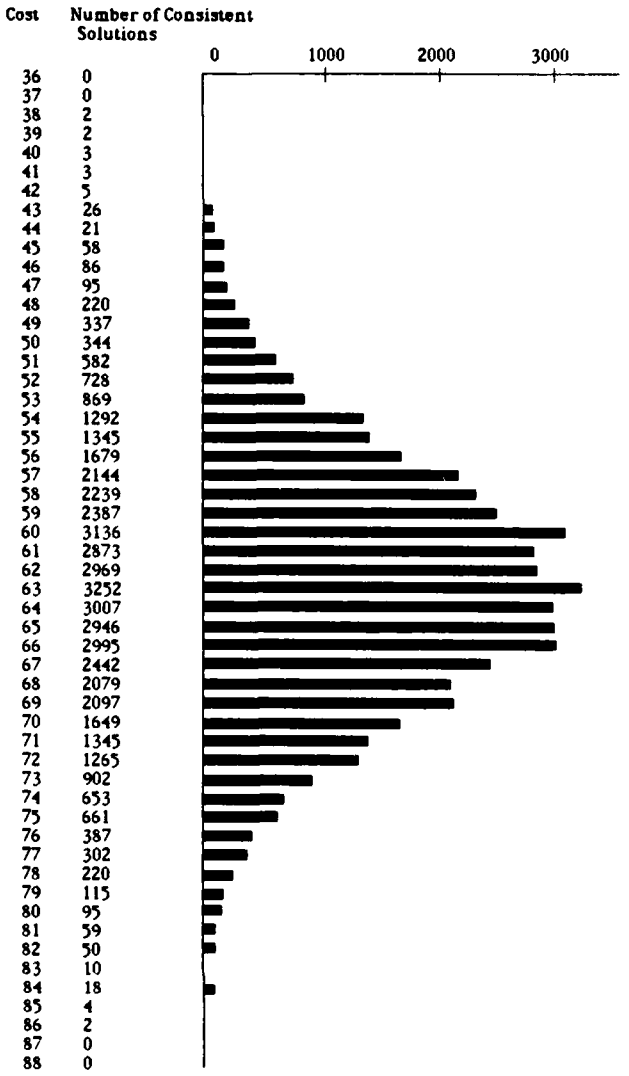


Fig. 4. Flow cost histogram for the Hitchcock problem. The number of samples is 50,000.

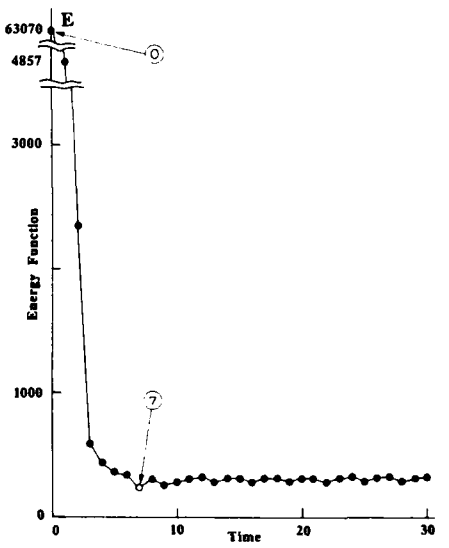


Fig. 5. Second example of the Hitchcock problem using a binary number representation scheme. Uniformly fuzzy states initialized the network, and a softer nonlinear function was used to give the best solution with a flow cost of 49. The weights used were $A = 27, B = C = 80, D = 0.2$. The constant x_0 was 1.0. The open circle represents a solution that satisfied the constraints. See Table IV for flow matrices at the iteration numbers indicated by the arrows.

Table IV. Flow Matrices for the Specified Numbers of Iterations Corresponding to Points Indicated on Fig. 5

No. 0		D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1	5.0	3.5	3.5	3.5	3.5	3.5
S2	3.0	3.5	3.5	3.5	3.5	3.5
S3	4.0	3.5	3.5	3.5	3.5	3.5
S4	6.0	3.5	3.5	3.5	3.5	3.5

(a)

No. 7		D1 2.0	D2 7.0	D3 3.0	D4 4.0	D5 4.0
S1	5.0	0	5.0	0	0	0
S2	3.0	2.0	0	1.0	0	0
S3	4.0	0	1.0	1.0	2.0	0
S4	6.0	0	1.0	1.0	0	4.0

(b)

be $A = 29, B = 80, C = 80$, and $D = 0.55$. Through the first several iterations, the source and demand constraints came to be almost satisfied (see Fig. 6 and Table V), and at the sixth iteration the first feasible solution, with cost 43, was reached (see Fig. 7 and Table VI). The solution was improved further by continuing iterations, passing another feasible solution with cost 40 at the tenth iteration; one of the best solutions with cost 38 was finally reached on the twenty-first iteration. To show the role played by the degeneracy of the number representation, the complete states of the 140 neurons are depicted in Fig. 8 for the iterations from 21 through 28. Each neuron is represented by a star when it is firing ($V_{XY,i} = 1$) and by a dot when not firing ($V_{XY,i} = 0$). The number of neurons that are firing in each set of seven neurons represents the content of the flow matrix element f_{XY} at the corresponding position. At iteration 21, for example, we had $f_{25} = 1$ because only one neuron $V_{25,3}$ was firing ($V_{25,3} = 1$), and the rest of the six neurons were not firing. At iteration 22, neuron $V_{25,3}$ stopped firing, but the correct solution $f_{25} = 1$ was retained because the next neighbor neuron $V_{25,2}$ started firing instead of $V_{25,3}$. We can observe a similar phenomenon in other sets of neurons representing f_{35} and f_{45} at iterations 21, 22, 23, 25, 26, and 27. In this manner, the neural network can give correct solutions at many different points in its state space, and these points cluster in a particular region of the state space that corresponds to low-energy function values. It is because of

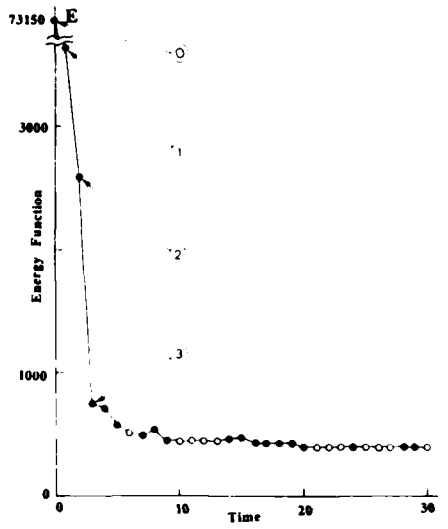


Fig. 6. Network dynamics of the Hitchcock problem using a degenerate (simple sum) number representation scheme. The constants used were $A = 29$, $B = C = 80$, $D = 0.55$, and $x_0 = 0.1$. Open circles again represent solutions that satisfy the constraints. Flow matrices corresponding to the arrows are found in Table V.

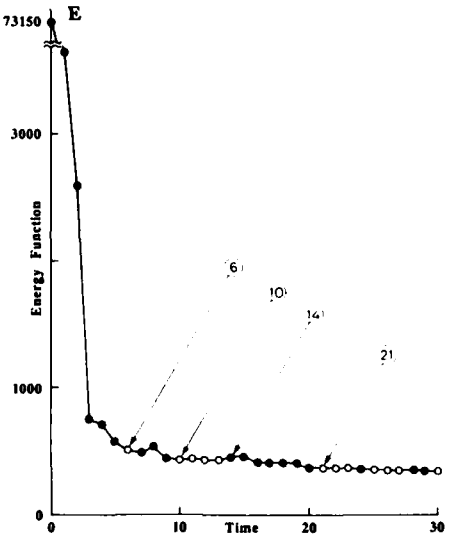


Fig. 7. Continuation of the example shown in Fig. 6. One of the two in 50,000 best solutions is found at time 21. Open circles represent solutions that satisfy the constraints (i.e., consistent solutions). The cost associated with the solution at the sixth iteration is 43, that associated with the group of consistent solutions starting at iteration 10 is 40, and that associated with the remaining consistent solutions is 38. See Table VI for the corresponding flow matrices.

this characteristic that the degenerate number representation scheme can have better problem-solving capabilities than the pure binary number representation scheme.

Figure 9 and Table VII show another example of the computational performance of the 140-neuron network, where uniform random numbers between 0 and 1 were assigned to the initial state variables of the neurons. In this example, we obtained two different solutions with cost 38, showing that the best solution is not unique.

Table V. Flow Matrices for the Specified Numbers of Iterations Corresponding to the Points Indicated on Fig. 6

No. 0	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	3.5	3.5	3.5	3.5	3.5
S2 3.0	3.5	3.5	3.5	3.5	3.5
S3 4.0	3.5	3.5	3.5	3.5	3.5
S4 6.0	3.5	3.5	3.5	3.5	3.5

(a)

No. 1	D1 2.0	D2 7.0	D3 3.0	D4 4.0	D5 4.0
S1 5.0	0	0	0	0	0
S2 3.0	0	0	0	0	0
S3 4.0	0	0	0	0	2.0
S4 6.0	0	1.0	0	2.0	2.0

(b)

No. 2	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	3.0	4.0	0	0	0
S2 3.0	1.0	2.0	1.0	0	0
S3 4.0	0	0	1.0	0	1.0
S4 6.0	0	1.0	1.0	1.0	2.0

(c)

No. 3	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	4.0	0	1.0	0
S2 3.0	1.0	1.0	0	0	0
S3 4.0	1.0	1.0	0	0	1.0
S4 6.0	0	1.0	2.0	1.0	2.0

(d)

V. Simultaneous Equations

In this section we show how we can program and solve on a neural network simultaneous equations

$$\mathbf{HX} = \mathbf{y}, \tag{30}$$

where $\mathbf{H}$ is a full-rank square matrix with $N \times N$ elements, and $\mathbf{x}$ and $\mathbf{y}$ are vectors with N elements representing, respectively, unknown and given variables. (Note that deconvolution is a special case of this general problem.)

Table VI. Flow Matrices for the Specified Numbers of Iterations Corresponding to the Points Indicated in Fig. 7

No. 6	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	4.0	0	1.0	0
S2 3.0	2.0	1.0	0	0	0
S3 4.0	0	1.0	0	1.0	2.0
S4 6.0	0	1.0	3.0	0	2.0

(a)

No. 10	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	4.0	0	0	1.0
S2 3.0	2.0	1.0	0	0	0
S3 4.0	0	1.0	0	1.0	2.0
S4 6.0	0	1.0	3.0	0	2.0

(b)

No. 14	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	4.0	0	0	1.0
S2 3.0	2.0	0	0	0	1.0
S3 4.0	0	0	0	3.0	0
S4 6.0	0	2.0	3.0	0	1.0

(c)

No. 21	D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1 5.0	0	5.0	0	0	0
S2 3.0	2.0	0	0	0	1.0
S3 4.0	0	0	0	2.0	2.0
S4 6.0	0	2.0	3.0	0	1.0

(d)

A. Energy Function

To use the spontaneous energy-minimization process of the neural network, we reformulate the problem in the form of a minimization problem by introducing an energy function that includes a term

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2, \tag{31}$$

so that the norm of the difference can be minimized through the energy minimization process. For our later demonstration of the Fourier transformation, we allow y and H to take on complex values, but, for the sake of simplicity, we restrict x to only positive integer values, although we could include complex numbers by using additional neurons labeled by a more complicated set of subscripts. As in Eq. (21), we express the nth element x_n of the unknown vector x by the group-and-weight scheme:

$$x_n = \sum_{k=1}^{K} \left[(M+1)^{k-1} \sum_{i=1}^{M} V_{n,(k-1)M+i} \right]. \tag{32}$$

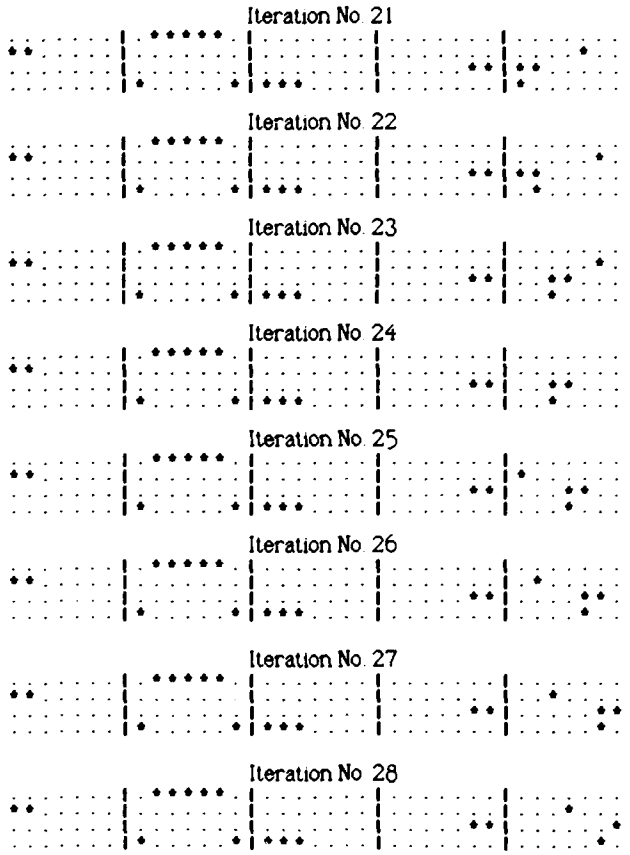


Fig. 8. Neural state transitions of the degenerate (simple sum) Hitchcock network (Figs. 6 and 7). Iterations 21–28 are shown.

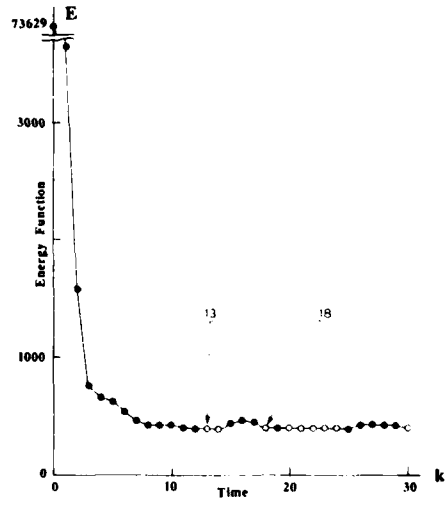


Fig. 9. Second example of the degenerate (simple sum) Hitchcock network. A random initial state drove this network to find both of the best solutions. The two-flow matrices are shown in Table VII.

Table VII. Flow Matrices for the Specified Numbers of Iterations Corresponding to the Points Indicated on Fig. 9

No. 13		D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1	5.0	0	4.0	0	0	1.0
S2	3.0	2.0	1.0	0	0	0
S3	4.0	0	0	0	2.0	2.0
S4	6.0	0	2.0	3.0	0	1.0

(a)

No. 18		D1 2.0	D2 7.0	D3 3.0	D4 2.0	D5 4.0
S1	5.0	0	5.0	0	0	0
S2	3.0	2.0	0	0	0	1.0
S3	4.0	0	0	0	2.0	2.0
S4	6.0	0	2.0	3.0	0	1.0

(b)

By substituting Eq. (32) into Eq. (31), we have an energy function

$$E = -(A/2)\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}[1-2V_{n,(k-1)M+i}]^2$$

$$+ (B/2)\sum_{l=1}^{N}\left(y_l - \sum_{n=1}^{N} h_{ln}x_n\right)\left(y_l^* - \sum_{n'=1}^{N} h_{ln'}^* x_{n'}\right)$$

$$= -(A/2)\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}[1-2V_{n,(k-1)M+i}]^2$$

$$+ (B/2)\sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{n'=1}^{N}\sum_{k=1}^{K}\sum_{k'=1}^{K}\sum_{i=1}^{M}\sum_{i'=1}^{M}(M+1)^{k+k'-2}h_{ln}h_{ln'}^*$$

$$\times V_{n,(k-1)M+i}V_{n',(k'-1)M+i'}$$

$$- B\sum_{l=1}^{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M}(M+1)^{k-1}\,\mathrm{Re}[y_l h_{ln}^*]V_{n,(k-1)M+i}$$

$$+ (B/2)\sum_{l=1}^{N}|y_l|^2, \tag{33}$$

where, as in Eq. (22), the first term is for binarization, y_l and h_{ln}, are elements of **y** and **H**, and * and Re[ ] denote complex conjugate and real part, respectively.

B. Interconnection Matrix

The energy function is now modified to

$$E = -(1/2)\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M}\sum_{n'=1}^{N}\sum_{k'=1}^{K}\sum_{i'=1}^{M}T_{n,(k-1)M+i;n',(k'-1)M+i'}$$

$$\times V_{n,(k-1)M+i}V_{n',(k'-1)M+i'}$$

$$- \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{i=1}^{M}V_{n,(k-1)M+i}I_{n,(k-1)M+i}, \tag{34}$$

By equating the corresponding coefficients of Eqs. (33) and (34), we determine the interconnection strengths and the biases:

$$T_{n,(k-1)M+i;n',(k'-1)M+i'} = 4A(M+1)^{k-1}\delta_{nn'}\delta_{kk'}\delta_{ii'}$$

$$- B(M+1)^{k+k'-2}\sum_{l=1}^{N}h_{ln}h_{ln'}^*, \tag{35}$$

$$I_{n,(k-1)M+i} = -2A(M+1)^{k-1} + B(M+1)^{k-1}$$

$$\times \mathrm{Re}\left[\sum_{l=1}^{N}h_{ln}^* y_l\right]. \tag{36}$$

Equation (31) includes the discrete Fourier transform as a special case with

$$h_{ln} = \exp[-2\pi j(l-1)(n-1)/N], \tag{37}$$

and the inverse transform is computed by solving the simultaneous linear equations.

In this case, Eq. (35) takes a simple form due to the orthogonality of the Fourier transform matrix:

$$T_{n,(k-1)M+i;n',(k'-1)M+i'} = 4A(M+1)^{k-1}\delta_{nn'}\delta_{kk'}\delta_{ii'}$$

$$- BN(M+1)^{k+k'-2}\delta_{nn'}. \tag{38}$$

C. Numerical Experiments

Computations of the inverse Fourier transform were programmed on the neural network, and the performance was simulated on a digital computer. We used signals with $N = 15$ sample points. Each sample point x_n was expressed by 24 neurons based on the simple sum scheme ($M = 24$, $K = 1$), so that 360 neurons were employed in total. We adopted the differential asynchronous transition mode and chose weight factors as $A = 28$ and $B = 1$. In Fig. 10, (a) and (b) show, respectively, an original signal **x** and its Fourier transform **y**. (Only absolute values are shown in the figure.) The task given to the neural network is to compute **x** from a given **y**.

Assuming no *a priori* knowledge, we started from the fuzziest initial states $V_{n,i} = 0.5$ shown in Fig. 10(c) and got the result shown in Fig. 10(d) after only two iterations. Another example is shown in Fig. 11, where we used an asymmetric signal and started from random initial states. Again after only two iterations we obtained the result shown in Fig. 11(d). Although the solutions obtained are not exact, the speed of computation is impressive. In fact, this apparently enormous speed of computation is quite misleading for reasons that will be revealed in the following section.

VI. Computational and Programming Complexities

As has been demonstrated in Secs. IV and V, the computational speed of a neural network is very high, solutions (although not always exact) being obtained within several clock times (iterations). At present, we do not know how the computation time (the number of iterations required) is related to the problem size (the number of neurons employed) and to the algorithm (the choice of the interconnections). We conjecture

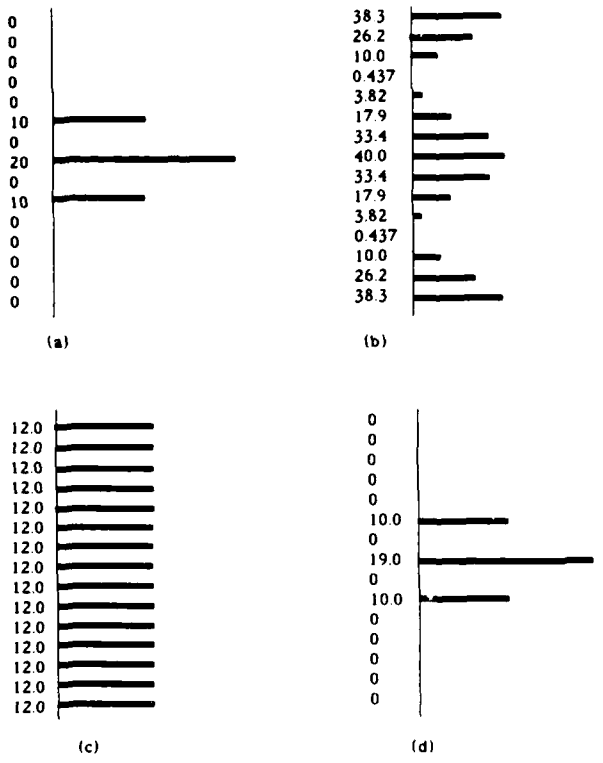


Fig. 10. Inverse DFT. The transition mode is differential asynchronous: (a) unknown signal; (b) known Fourier transform; (c) uniformly fuzzy initial states; (d) estimated signal after two iterations.

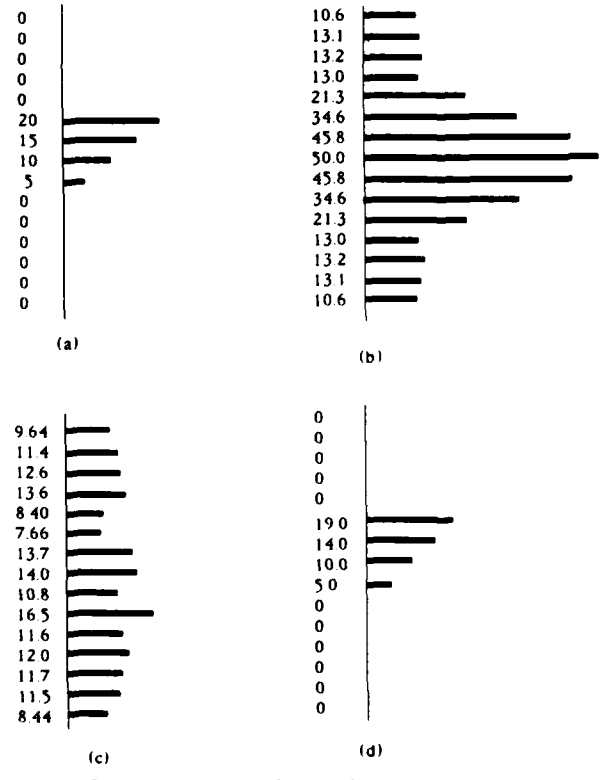


Fig. 11. Inverst DFT, second example: (a) unknown asymmetric signal; (b) known Fourier transform; (c) random initial states; (d) estimated signal after two iterations.

that the computation time does not grow too rapidly with problem size, because the greater the problem size, the more neurons participate in solving the problem and the higher the parallelism used. If this conjecture is correct, the computation time is very short for a properly programmed (interconnected) neural network, irrespective of the problem size. It may appear, then, that neural networks would be the computation architecture of choice in most problems that can be included within the energy minimization framework. However, this conclusion is not correct. Although the computation time itself may be very short, it may be necessary to invest very significant computation time simply to program the network, i.e., to determine the proper interconnection strengths and neural biases. The situation is somewhat analogous to the classical analog electronic computer for which a large amount of time must be spent wiring the proper modules together before any problem can be solved. Once the modules are connected, a solution appears almost immediately.

A. Programming Complexity

By analogy with the concept of computational complexity[14,15] in digital computing, we introduce the concept of programming complexity in neural computing. We define programming complexity as the number of arithmetic operations that must be performed to determine the proper interconnection strengths and neural biases for the problem to be solved. Conventional digital computers also need programming, but once the program is compiled and stored in memory, it can be used on many different sets of input data. For this reason, the concept of programming complexity has little significance in the world of conventional digital computers, where programs are completely separable from data. In neural network computers, a program and data are generally mixed and stored in the interconnection strengths and/or neural biases. For example, in Eq. (24), the first three terms represent part of the program (since they do not depend on data), and the last term, including the costs C_{XY}, corresponds to the data. Therefore, we must redetermine the interconnection strengths and/or the biases each time we use new data. In such an environment, the programming complexity becomes an important measure of the efficiency of neural computing. We know that it is not meaningful to compare the efficiencies of conventional digital computers and neural computers on the basis of computational complexity and programming complexity, because they mean different things. Digital computers always give exact solutions (within the machine precision) after performing the number of operations specified by the computational complexity, whereas neural computers do not guarantee exact solutions even if they are programmed by performing the number of operations specified by the programming complexity. Nevertheless, a comparison of the computational complexity and the programming complexity does reveal certain interesting aspects of neural computing, as discussed in the following section.

B. Simultaneous Equations

To solve simultaneous equations with N unknown variables, we employed qN neurons, with q being the number of neurons used to represent each unknown variable. We consider q to be a constant factor, since it does not depend on N. The number of interconnections is given by $(1/2)qN(qN + 1) \approx O(N^2)$, and the number of biases is $qN \approx O(N)$. We need $O(N)$ operations to determine each interconnection strength [see Eq. (35)] and each bias [see Eq. (36)], so that the programming complexity is $O(N^3)$. The computational complexity of this problem is also $O(N^3)$.[14] This means that solutions of such a problem on either a neural computer or a conventional digital computer would require essentially the same computational load. In the case of the neural computer, the computations must be expended to determine the interconnection strengths and biases, while in the case of the conventional digital computer the computations are expended on solving the problem itself.

This comparison is even more striking in the case of the Fourier transformation discussed earlier. Since Eq. (38) contains no data terms, we need not recompute the interconnection strengths for each different set of data. The programming complexity comes only from computation of the term

$$\sum_{l=1}^{N} h^{*}_{ln} y_l$$

in the biases, Eq. (36). Noting Eq. (37), we find that to determine the proper biases, we must in fact compute the very same inverse Fourier transform that the neural network was to find! Thus we have already arrived at the solution by the time we finish programming, and it is now no surprise the neural network supplies the answer in only two interactions. The answer is in fact preprogrammed into the machine!

C. Traveling Salesman Problem

In the previous section we saw an example in which the programming complexity of a neural computer and the computational complexity on a conventional computer are of the same order. The question naturally arises as to whether this is the case with all problems. If so, neural computing loses most of its attractiveness. Hopfield and Tank's paper[13] on the traveling salesman problem provides the best example with which to answer this question. The computational complexity of the traveling salesman problem is an exponential function $O(N!)$ of the number of cities N. Hopfield and Tank showed that the problem can be programmed on a neural network with N^2 neurons that represent the elements of a permutation matrix. We can show that the programming complexity of this scheme is $O(N^3)$. This large difference of complexities makes neural computing very attractive, even though it does not guarantee the best solution.

C. Hitchcock Problem

Computational complexity in conventional digital computing depends greatly on the algorithms used, so that a great effort has been made by computer scientists to seek better algorithms and thereby reduce computational complexity. The same can be true with programming complexity in neural computing. The Hitchcock problem provides a good example for demonstrating good and poor algorithms (ways of interconnection) in terms of programming complexity. In Sec. IV, the Hitchcock problem with m sources and n demands was solved by using $qmn \approx O(mn)$ neurons. Since Eqs. (24) and (25) include data C_{XY}, S_X, and D_Y, we have to redetermine $(1/2)qmn(1 + qmn) \approx O(m^2n^2)$ interconnection strengths and $qmn \approx O(mn)$ biases for each new set of data. Each interconnection strength and bias can be determined by a constant number of operations, so that the programming complexity is given by $O(m^2n^2) = O(n^4)$ for $m \approx n$. In Sec. IV.B, we suggested an alternative definition of the energy function that does not square the total cost in the last term of Eq. (22). If we use this new energy function, the interconnection strengths and biases become

$$\begin{aligned} T_{XY,(k-1)M+i;X'Y',(k'-1)M+i'} = {} & 4A(M+1)^{k-1}\delta_{XX'}\delta_{YY'}\delta_{kk'}\delta_{ii'} \\ & - B(M+1)^{k+k'-2}\delta_{XX'} \\ & - C(M+1)^{k+k'-2}\delta_{YY'}, \end{aligned} \tag{39}$$

$$\begin{aligned} I_{XY,(k-1)M+i} = {} & -2A(M+1)^{k-1} + B(M+1)^{k-1}S_X \\ & + C(M+1)^{k-1}D_Y \\ & - (1/2)D(M+1)^{k-1}C_{XY}. \end{aligned} \tag{40}$$

Now the interconnection strengths do not depend on the data C_{XY}, and they need not be redetermined for each new set of data, so that the programming complexity comes only from the biases, Eq. (40), and is given by $O(mn) \approx O(n^2)$ for $m \approx n$. This is a very significant improvement. The computational complexity of the Hitchcock problem depends on the algorithm used by a conventional digital computer. If we search for the best solution randomly among all the possible combinations of the neural states, it becomes $2^{qmn} \approx O(2^{mn})$. Even if we restrict the search to feasible solutions, it can still be exponential $O(n^{m-1}m^{n-1})$.[18] Of course, these algorithms are worst extremes, and there exist several good algorithms that are in practical use. We do not know exactly what is the computational complexity of the best existing algorithm for the Hitchcock problem, but we estimate it to be a low-order polynomial. If it is still higher than $O(mn)$, neural computing can have an advantage for this problem.

VII. Conclusion

Following the lead of Hopfield and Tank, we proposed an architecture for programming highly parallel computation on neural networks. In Sec. III, we described number representation schemes based on linear mapping of the number space onto the neuron space and pointed out the advantage of the degenerate number representation schemes. In Secs. IV and V, the validity of the architecture was demonstrated by solving the Hitchcock problem and simultaneous linear equations on neural networks. The dynamics of the neural network were simulated on a digital com-

puter. In Sec. VI, we introduced the new concept of programming complexity in neural computing, which was used to evaluate the computational efficiency of algorithms performed on neural networks. We compared the programming complexity with the worst case computational complexity, simply because the average complexity was too hard to estimate. However, we note that programming complexity is better compared with average computational complexity, because they have a common characteristic that the solution is not always best or exact, even if we perform the number of operations specified by these complexities.

Finally, we point out that there exists a fundamental limitation to the class of problem that can be programmed and solved on the Hopfield neural network. This limitation comes from the requirement that the energy function must be a quadratic function of the neuron state variables. All linear problems, such as discussed in this paper, can satisfy this requirement. However, general nonlinear problems cannot satisfy this requirement. Floating-point number representation is one such nonlinear problem.

We wish to thank Dorothy Mighell for careful reading of the manuscript on numerous occasions and many useful suggestions for improvement.

When this work was done, Mitsuo Takeda was on scholarly leave from the University of Electrocommunications, Tokyo, Japan.

References

1. See, for exampl, G. E. Hinton, "Learning in Parallel Networks," Byte **10,** 265 (1985); J. A. Feldman, "Connections," Byte **10,** 277 (1985), a special issue for artificial intelligence.
2. See, for example, J. J. Hopfield, "Brain, Computer, and Memory," Eng. Sci. **46,** 2 (1982).
3. J. W. Goodman, F. J. Leonberger, S.-Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE **72,** 850 (1984).
4. D. Psaltis and N. Farhat, "Optical Information Processing Based on an Associative-Memory Model of Neural Nets with Thresholding and Feedback," Opt. Lett. **10,** 98 (1985).
5. G. Eichmann and H. J. Caulfield, "Optical Learning (Inference) Machines," Appl. Opt. **24,** 2051 (1985).
6. H. Mada, "Architecture for Optical Computing Using Holographic Associative Memories," Appl. Opt. **24,** 2063 (1985).
7. N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical Implementation of the Hopfield Model," Appl. Opt. **24,** 1469 (1985).
8. J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Natl. Acad. Sci. **79,** 2554 (1982).
9. J. J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-state Neurons," Proc. Natl. Acad. Sci. **81,** 3088 (1984).
10. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully-Parallel High-Speed Incoherent Optical Method for Performing Discrete Fourier Transforms," Opt. Lett. **2,** 1 (1978).
11. See, for example, S. Amari, *Mathematical Theory of Neural Networks* (Sangyotosho, Tokyo, 1978), in Japanese; T. Kohonen, *Self-Organization and Associative Memory* (Springer-Verlag, New York, 1984).
12. See, for example, T. Kohonen, *Content Addressable Memories* (Springer-Verlag, New York, 1980); K. Nakano, "Associatron—A Model of Associative Memory," IEEE Trans. Syst. Man Cybern. **SMC-2,** 380 (1972).
13. J. J. Hopfield and D. W. Tank, "Neural Computation of Decisions in Optimization Problems," Biocybernetics Jena (1985), to appear.
14. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA, 1974).
15. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Prentice-Hall, Englewood Cliffs, NJ, 1982); K. Murty, *Linear and Combinatorial Programming* (Wiley, New York, 1976).
16. M. Iri, *Network Flow, Transportation and Scheduling* (Academic, New York, 1969); D. T. Phillips and A. Garcia-Dias, *Fundamentals of Network Analysis* (Prentice-Hall, New York, 1981).
17. F. L. Hitchcock, "The Distribution of a Product from Several Sources to Numerous Localities," J. Math. Phys. **20,** 224 (1941).
18. S. I. Glass, *Linear Programming* (McGraw-Hill, New York, 1958).

A Generalized Convergence Theorem for Neural Networks and its Applications in Combinatorial Optimization

Jehoshua Bruck
Joseph W. Goodman
Department of Electrical Engineering
Information Systems Laboratory
Durand Building, Stanford University
Stanford, CA 94305

This paper deals with a neural network model in which each neuron performs a threshold logic function. An important property of the model is that it always converges to a stable state when operating in a serial mode and to a cycle of length at most 2 when operating in a fully parallel mode [3,4]. This property is the basis of the potential applications of the model, such as associative memory devices and combinatorial optimization [5].

The paper reviews the two known convergence theorems (for serial and fully parallel modes of operation) and presents a general convergence theorem which unifies the two known cases. The paper also presents some new applications of the model for combinatorial optimization. In particular, new relations between the neural network model and the problem of finding the Minimum Cut in graph are presented.

1 Background

The neural network model is a discrete time system that can be represented by a weighted and undirected graph. There is a weight attached to each edge of the graph and a threshold value attached to each node (neuron) of the graph. The *order* of the network is the number of nodes in the corresponding graph. Let N be a neural network of order n; then N is uniquely defined by (W,T) where:

- W is an $n \times n$ symmetric matrix, where W_{ij} is equal to the weight attached to edge (i,j).
- T is a vector of dimension n, where T_i denotes the threshold attached to node i.

Every node (neuron) can be in one of two possible states, either 1 or -1. The state of node i at time t is denoted by $V_i(t)$. The *state* of the neural network at time t is the vector $V(t)$.

The next state of a node is computed by:

$$V_i(t+1) = sgn(H_i(t)) = \begin{cases} 1 & \text{if } H_i(t) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

where

$$H_i(t) = \sum_{j=1}^{n} W_{j,i} V_j(t) - T_i$$

The next state of the network, i.e. $V(t+1)$, is computed from the current state by performing the evaluation (1) at a set S of the nodes of the network. The modes of operation are determined by the method by which the set S is selected in each time interval. If the computation is performed at a single node in any time interval, i.e. $\mid S \mid = 1$, then we will say that the network is operating in a *serial* mode, and if $\mid S \mid = n$ then we will say that that the network is operating in a *fully parallel* mode. All the other cases, i.e. $1 <\mid S \mid < n$ will be called *parallel* modes of operation. The set S can be chosen at random or according to some deterministic rule.

A state $V(t)$ is called *stable* iff $V(t) = sgn(WV(t) - T)$, i.e. there is no change in the state of the network no matter what the mode of operation.

2 Convergence Theorems

One of the most important properties of the model is the fact that it always converges, as summarized by the following theorem.

Theorem 1 *Let $N = (W, T)$ be a neural network, with W being a symmetric matrix then:*

1. *(Hopfield [4]) If N is operating in a serial mode and the elements of the diagonal of W are nonnegative then the network will always converge to a stable state, i.e. there are no cycles in the state space.*
2. *(Goles [3]) If N is operating in a fully parallel mode then the network will always converge to a stable state or to a cycle of length 2, i.e. the cycles in the state space are of length ≤ 2.*

The main idea in the proof of the two parts of the theorem is to define a so called *energy function* and to show that this energy function is nondecreasing when the state of the network changes. Since the energy function is bounded from above it follows that the energy will converge to some value. An important note is that originally the energy function was defined such that it is nonincreasing [3,4]; we changed it to be nondecreasing such that the value of the energy will comply with some known graph problems (e.g. Min Cut, see next section).

The second step in the proof is to show that constant energy implies in the first case a stable state, and in the second a cycle of length ≤ 2. The energy functions defined for each part of the proof are different,

$$\begin{aligned} E_1(t) &= V^T(t)WV(t) - (V(t) + V(t))^T T \\ E_2(t) &= V^T(t)WV(t-1) - (V(t) + V(t-1))^T T \end{aligned} \tag{2}$$

where $E_1(t)$ and $E_2(t)$ denote the energy functions related to the first and second part of the proof.

An interesting question is whether two different energy functions are needed in order to prove the two parts of theorem 1. A new result is that convergence in the fully parallel mode can be proven using the result on convergence for the serial mode of operation.

The following lemma will describe a general result which enables transformation of a neural network with nonnegative self loops operating in a serial mode to an equivalent network without self loops (part a), and also enables transformation of a neural network operating in a fully parallel mode to an equivalent network operating in a serial mode (part b). The equivalence is in the sense that it is possible to derive the state of one network given the state of the other network, provided the two networks started from the same initial state.

Lemma 1 *Let $N = (W, T)$ be a neural network.*
Let $\hat{N} = (\hat{W}, \hat{T})$ be obtained from N as follows:
$\hat{N}$ is a bipartite graph, with $\hat{W} = \begin{pmatrix} 0 & W \\ W & 0 \end{pmatrix}$ and $\hat{T} = \begin{pmatrix} T \\ T \end{pmatrix}$
Claims:

(a) *For any serial mode of operation in N there exists an equivalent serial mode of operation in $\hat{N}$; provided W has a nonnegative diagonal.*

(b) *There exists a serial mode of operation in $\hat{N}$ which is equivalent to a fully parallel mode of operation in N.*

Proof: The new network $\hat{N}$ is a bipartite graph with $2n$ nodes, the set of nodes of $\hat{N}$ can be subdivided into two sets: let P_1 and P_2 denote the set of the first and the last n nodes, respectivelv. Clearly, no two nodes of P_1 (and also P_2) are connected by an edge; that is, both P_1 and P_2 are independent sets of nodes in $\hat{N}$ (an independent set of nodes in a graph is a set of nodes in which no two nodes are connected by an edge). Another observation is that P_1 and P_2 are symmetric in the sense that a node $i \in P_1$ has an identical edge set as has a node $(i + n) \in P_2$.

Proof of (a): Let V_0 be an initial state of N, and let $(i_1, i_2...)$ be the order by which the states of the nodes are evaluated in a serial mode in N. We will show that starting from the initial state (V_0, V_0) in $\hat{N}$ (the state of both P_1 and P_2 is V_0) and using the order $(i_1, (i_1 + n), i_2, (i_2 + n), ...)$ for the evaluation of states will result in:

1. The state of P_1 will be equal to the state of P_2 in $\hat{N}$ after an arbitrary even number of evaluations.

2. The state of N at time k is equal to the state of P_1 at time $2k$, for an arbitrary k.

The proof of (1) is by induction. Given that at some arbitrary time k the state of P_1 is equal to the state of P_2, it will be shown that after performing the evaluation at node i and then at node $(n + i)$ the states of P_1 and P_2 remain equal.

There are two cases:

- If the state of node i does not change as a result of evaluation, then by the symmetry of $\hat{N}$ there will be no a change in the state of node $(n + i)$.

- If there is a change in the state of node i, then because $\hat{W}_{i,n+i}$ is nonnegative it follows that there will be a change in the state of node $(n + i)$ (the proof is straightforward and won't be presented).

The proof of (2) follows from (1): by (1) the state of P_1 is equal to the state of P_2 right before the evaluation at a node of P_1. The proof is by induction: assume that the current state of N is the same as the state of P_1 in $\hat{N}$. Then an evaluation performed at a node $i \in P_1$ will have the same result as an evaluation performed at node $i \in N$. □

Proof of (b): Let's assume as in part (a) that $\hat{N}$ has the initial state (V_0, V_0). Clearly, performing the evaluation at all nodes belonging to P_1 (in parallel) and then at all nodes belonging to P_2, and continuing with this alternating order is equivalent to a fully parallel mode of operation in N. The equivalence is in the sense that the state of N is equal to the state of the subset of nodes (either P_1 or P_2) of $\hat{N}$ at which the last evaluation was performed. A key observation is that P_1 and P_2 are independent sets of nodes, and a parallel evaluation at an independent set of nodes is equivalent to a serial evaluation of all the nodes in the set [1]. Thus, the fully parallel mode of operation in N is equivalent to a serial mode of operation in $\hat{N}$. □

Using the transformations suggested by the above lemma it is possible to explore some of the relations between convergence properties as summarized by the following theorem.

Theorem 2 *Let $N = (W,T)$ be a neural network.*
Then (2) and (3) are implied by (1).

1. *If N is operating in a serial mode and W is a symmetric matrix with zero diagonal, then the network will always converge to a stable state.*
2. *If N is operating in a serial mode and W is a symmetric matrix with nonnegative elements on the diagonal, then the network will always converge to a stable state.*
3. *If N is operating in a fully parallel mode then, for an arbitrary symmetric matrix W, the network will always converge to a stable state or a cycle of length 2; that is, the cycles in the state space are of length ≤ 2.*

Proof: The proof is based on lemma 1.

(2) is implied by (1): by lemma 1 part (a) every neural network with nonnegative diagonal matrix W which is operating in a serial mode can be transformed to an equivalent network to be denoted by $\hat{N}$ which is operating in a serial mode with W being a zero diagonal matrix. $\hat{N}$ will converge to a stable state (by (1)); hence, N will also converge to a stable state which will be equal to the state of P_1. Note that trivially (1) is implied by (2). □

(3) is implied by (1): by lemma 1 part (b) every neural network operating in a fully parallel mode can be transformed to an equivalent neural network to be denoted by $\hat{N}$ which is operating in a serial mode and with $\hat{W}$ being a zero diagonal matrix. $\hat{N}$ will converge to a stable state (by (1)). When $\hat{N}$ reachs a stable state there are two cases:

1. The state of P_1 is equal to the state of P_2; in this case N will converge to a stable state which is equal to the state of P_1.
2. The states of P_1 and P_2 are distinct; in this case N will oscillate between the two states defined by P_1 and P_2, i.e. N will converge to a cycle of length 2. □

It is also interesting to investigate the relations between the two energy functions in a neural network operating in a fully parallel mode or in a serial mode. New results concerning this question are summarized in the following theorem.

Theorem 3 *Let $N = (W,T)$ be a neural network. Then:*

(a) *For N operating in a serial mode, and for all t:*
$E_1(t-1) \leq E_2(t) \leq E_1(t)$

(b) *For N operating in fully parallel mode, and for all t:*
$E_2(t) \geq E_1(t-1)$
$E_2(t) \geq E_1(t)$ *when the network is in a cycle of length two.*
$E_2(t) \leq E_1(t)$ *when the network is in a stable state.*

Proof: The proof of theorem 3 can be done by straightforward algebraic operations. It turns out that theorem 3 can also be proven by using Lemma 1 and the fact that the energy E_1 is nondecreasing in a network operating in a serial mode (Theorem 1a). We include a sketch of the alternative proof to emphasize the power of Lemma 1 for understanding the relations between the two energy functions and the two modes of operation. In the following proofs we will use the notations established in Lemma 1.

- Proof of part a:
  Perform the transformation of N to $\hat{N}$; there is a way to simulate a serial operation in N by a serial operation in $\hat{N}$ (as suggested by Lemma 1a) provided that W is a nonnegative diagonal matrix.

  Look at the energy E_1 of $\hat{N}$ to be denoted by $\hat{E}_1$. By Theorem 1a:

  $E_1(t) = \hat{E}_1(2t)$
  Also,

  $E_2(t+1) = \hat{E}_1(2t+1)$

  Since $\hat{N}$ is operating in a serial mode it follows that $\hat{E}_1$ is nondecreasing. □

- Proof of part b:
  The key idea in the proof is the simple observation that if a state with energy $\hat{E}_1(t+k)$ can be reached from a state with energy $\hat{E}_1(t)$ in k serial iterations; then it follows that $\hat{E}_1(t) \leq \hat{E}_1(t+k)$. If P_1 and P_2 in $\hat{N}$ have the same state as N at time t then $\hat{E}_1(t) = E_1(t)$. Clearly, performing one parallel iteration in N and on P_1 in $\hat{N}$ will result in $E_2(t+1) = \hat{E}_1(t+1)$. Hence, $E_2(t+1) \geq E_1(t)$ for every value of t when N is operating in a parallel mode.
  If N is in a cycle of length 2 then $\hat{E}_1(t) = E_2(t)$; by using the same arguments as above it follows that $E_2(t) \geq E_1(t)$.
  If N enters a stable state at time t, then $\hat{E}_1(t-1) = E_2(t)$ and also $\hat{E}_1(t) = E_1(t)$; thus, it follows that $E_2(t) \leq E_1(t)$. □

Some remarks regarding the above analysis:

1. In a network operating in a serial mode, both E_1 and E_2 are nondecreasing. Furthermore, a very interesting result (theorem (3) part (a)) is that the values of E_1 and E_2 are interleaving.

2. The assumption of W being a nonnegative diagonal matrix is used to derive results for a network operating in a serial mode only.

3. In a network operating in a fully parallel mode E_1 is not necessarily nondecreasing; it can be shown that a sufficient condition for E_1 to be nondecreasing is that W is nonnegative definite over the range (-1,0,1).

3 Application to Combinatorial Optimization

Theorem 1a implies that a neural network, when operating in a serial mode, will always get to a stable state which corresponds to a local maximum in the energy function E_1. This property suggests the use of the network as a device for performing a local search algorithm for finding a maximal value of the energy function E_1 [5]. The value of E_1 which corresponds to the initial state is improved by performing a sequence of random serial iterations until the network reaches a local maximum.

From Theorem 1b it follows that when the network is operating in a fully parallel mode it will always reach a stable state or a cycle of length 2. The value of the energy E_2 at these final points is clearly maximal with respect to the path in the state space which ends in these points. In the fully parallel case there is no randomness in the search, because there is no choice in the direction of improvement as in a serial operation. Actually, a network operating in a fully parallel mode is performing a deterministic mapping from the set of initial states to the set of final states (stable states and cycles of length 2). A random local search can be performed by using the construction suggested by Lemma 1.

One of the advantages of the construction suggested by Lemma 1 is that the state space of a network operating in a fully parallel mode can be transformed from a forest like graph to a $2n$-cube. This transformation enables the use of the network for performing a random and local search for the maximum of a function of the form of E_2.

To summarize, given a quadratic function of the form E_1 or E_2, it is possible to construct a neural network which will perform a **random** local search for the maximum.
The class of optimization problems which can be represented by quadratic functions is very rich. One of the problems which is not only representable by a quadratic function but actually is equivalent to it is the Min Cut problem [1,6].

In the sequel, we will present (without proof) the equivalence between the Min Cut problem and neural networks (Theorems 4 and 5), and also show how neural networks relate to the Directed Min Cut problem (Theorem 6). In order to make the above statements clear, let us start by defining the term cut in a graph.

Definition: Let $G = (V, E)$ be a weighted and undirected graph, with W being an $n \times n$ symmetric matrix of weights of the edges of G. Let V_1 be a non empty subset of V, and let $V_{-1} = V - V_1$. A *set of edges* each of which is incident at one node in V_1 and at one node in V_{-1} is called a *cut* of the graph G. The *Min Cut* of a graph is the cut for which the sum of the corresponding edge weights is minimal.

Theorem 4 *Let $N = (W, T)$ be a neural network with W being an $n \times n$ zero diagonal matrix.*
Let G be a weighted graph with $(n+1)$ nodes, with its weight matrix W_G being:

$$W_G = \begin{pmatrix} W & T \\ T^T & 0 \end{pmatrix}$$

The problem of finding the state V in N for which E_1 is maximum is equivalent to finding the Min Cut of the corresponding graph G.

The above theorem can also be generalized to show that the difference between energy values corresponding to two different states of a network is also equal to a cut in a graph. The following theorem summarizes this interesting relation.

Theorem 5 *Let $N = (W, T)$ be a neural network with $T = 0$. Let V_1 and V_2 be two arbitrary states of N. Let $N(V_1)$ be a network obtained from N by modifying W as follows:*

$$\textbf{modified } W_{i,j} = W_{i,j} V_{1,i} V_{1,j} \tag{3}$$

Then:

$E_1(V_1) - E_1(V_2) = 2 \times (\textbf{cut of } N(V_1))$

A few remarks concerning the above relation:

1. Theorem 4 is a special case of theorem 5; simply choose V_1 to be the all-1 vector.
2. Theorem 5 suggests an iterative method for improving the value of the current local maximum as follows:
   (a) Perform the search until reaching a local maximum.
   (b) Modify the weights of the network according to (3), go to (a).
   Clearly, we will get an improvement in each iteration as long as the new maximal value of the energy function is positive.

The Min Cut problem is known to be NP-hard [2]. The problem is solvable in polynomial time (by flow techniques) if the weights of the graph are nonnegative, and also there is a set of special cases (e.g. planar graphs) for which a polynomial algorithm is known.

The importance of the relation between the Min Cut problem and neural networks is in the fact that the Min Cut problem can be viewed as a generic graph problem which can be mapped to the model. Thus, theoretically one can transform every NP-complete problem to the Min Cut problem and use the corresponding neural network to perform a local search algorithm.

The relation to the Min Cut problem also leads to the following nice interpretation of a serial iteration in a neural network. The computation performed at a node k is equivalent to deciding what will be the next position of node k with respect to the current cut (state) of the network. The decision is performed by comparing the sum of weights of the edges which belong to the cut and incident at node k with the sum of weights of the other edges which are incident at node k.

In each serial iteration the value of the cut goes down; thus, the corresponding energy value goes up. This is actually a simple proof for the convergence in the serial mode of operation.

From the above equivalence it follows that the Min Cut problem in an undirected graph is trivially mapped to a neural network. What about directed graphs: is it possible to design a neural network which will search for a solution for the Directed Min Cut (DMC) problem [1]? It is shown in Theorem 6 that it is possible to map the DMC problem to a neural network which will perform a local search.

Definition: Let $G = (V, E)$ be a weighted and directed graph. Each edge has a direction and a weight. The weights of the directed edges (arcs) can be represented by an $n \times n$ matrix W in which $W_{i,j}$ is the weight of the arc from i to j. Let V_1 be a non empty subset of V, and let $V_{-1} = V - V_1$. The *set of arcs* each of which has its tail at a node in V_1 and its head at a node in V_{-1} is called a *directed cut* of G.

Theorem 6 *[1] Let $G = (V, E)$ be a weighted directed graph with W being the matrix of its edge weights (W is not necessarily symmetric). The network $N = (\tilde{W}, T)$ performs a local search for the Directed Min Cut of G where:*

$$\tilde{W}_{ij} = \frac{1}{2}(W_{ij} + W_{ji})$$

$$T_k = \frac{1}{2}\sum_{i=1}^{n}(W_{ki} - W_{ik})$$

Acknowledgement: Support of the Rothschild Foundation and the U.S. Air Force Office of Scientific Research are gratefully acknowledged.

References

[1] J. Bruck and J. Sanz, *A Study on Neural Network*, IBM ARC, Computer Science, RJ 5403, 1986.

[2] M. R. Garey and D.S. Johnson, *Computers and intractability, a Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, 1979.

[3] E. Goles, F. Fogelman and D. Pellegrin, *Decreasing Energy Functions as a Tool for Studying Threshold Networks*, Disc. Appl. Math. 12, pp. 261-277, 1985.

[4] J. J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Nat. Acad. Sci. . USA, Vol. 79, pp. 2554-2558, 1982.

[5] J. J. Hopfield and D. W. Tank, *Neural Computations of Decisions in Optimization Problems*, Biol. Cybern. 52, pp. 141-152, 1985.

[6] J. C. Picard and H. D. Ratliff, *Minimum Cuts and Related Problems*, Networks, Vol 5, pp. 357-370, 1974.

On the Power of Neural Networks for Solving Hard Problems

Jehoshua Bruck
Joseph W. Goodman
Department of Electrical Engineering
Information Systems Laboratory
Durand Building, Stanford University
Stanford, CA 94305

This paper deals with a neural network model in which each neuron performs a threshold logic function. An important property of the model is that it always converges to a stable state when operating in a serial mode [1,4]. This property is the basis of the potential applications of the model such as associative memory devices and combinatorial optimization [2,5].

One of the motivations for use of the model for solving hard combinatorial problems is the fact that it can be implemented by optical devices and thus operate at a higher speed than conventional electronics.

The main theme in this work is to investigate the power of the model for solving NP-hard problems [3], and to understand the relation between speed of operation and the size of a neural network. In particular, it will be shown that:

1. A network with polynomial (in the size of the input) number of neurons can not solve an NP-hard problem even if it operates for an exponential length of time (unless NP = co-NP).

2. A network with polynomial (in the size of the input) number of neurons which always gets to an ϵ-approximate solution for the Traveling Salesman Problem (TSP) [3,5] does not exist unless P=NP.

The above results are of great practical interest, because right now it is possible to build neural networks which will operate fast but are limited in the number of neurons.

1 Background

The neural network model is a discrete time system that can be represented by a weighted and undirected graph. There is a weight attached to each edge of the graph and a threshold value attached to each node (neuron) of the graph. The *order* of the network is the number of nodes in the corresponding graph. Let N be a neural network of order n; then N is uniquely defined by (W,T) where:

- W is an $n \times n$ symmetric matrix, where W_{ij} is equal to the weight attached to edge (i,j).
- T is a vector of dimension n, where T_i denotes the threshold attached to node i.

Every node (neuron) can be in one of two possible states, either 1 or -1. The state of node i at time t is denoted by $V_i(t)$. The *state* of the neural network at time t is the vector $V(t)$.

The next state of a node is computed by:

$$V_i(t+1) = sgn(H_i(t)) = \begin{cases} 1 & \text{if } H_i(t) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

where

$$H_i(t) = \sum_{j=1}^{n} W_{ji} V_j(t) - T_i$$

The next state of the network, i.e. $V(t+1)$, is computed from the current state by performing the evaluation (1) at a set S of the nodes of the network. The modes of operation are determined by the method by which the set S is selected in each time interval. If the computation is performed at a single node in any time interval, i.e. $\mid S \mid = 1$, then we will say that the network is operating in a *serial* mode, and if $\mid S \mid = n$ then we will say that that the network is operating in a *fully parallel* mode. All the other cases, i.e. $1 < \mid S \mid < n$ will be called *parallel* modes of operation. The set S can be chosen at random or according to some deterministic rule.

A state $V(t)$ is called *stable* iff $V(t) = sgn(WV(t) - T)$, i.e. there is no change in the state of the network no matter what the mode of operation. One of the most important properties of the model is the fact that it always converges to a stable state while operating in a serial mode. The main idea in the proof of the convergence property is to define a so called *energy function* and to show that this energy function is nondecreasing when the state of the network changes. The energy function is:

$$E(t) = V^T(t)WV(t) - 2V^T(t)T \quad (2)$$

An important note is that originally the energy function was defined such that it is nonincreasing [4]; we changed it such that it will comply with some known graph problems (e.g. Min Cut).

A neural network will always get to a stable state which corresponds to a local maximum in the energy function. This suggests the use of the network as a device for performing a local search algorithm for finding a maximal value of the energy function [5]. Thus, the network will perform a local search by operating in a random and serial mode. It is also known [1,7] that maximization of E associated with a given network N in which $T = 0$ is equivalent to finding the Minimum Cut in N. Actually, many hard problems can be formulated as maximization of a quadratic form (e.g. TSP [5]) and thus can be mapped to a neural network.

2 The Main Results

The set of stable states is the set of final solutions that one will get using the above approach. These final solutions correspond to local optima in the corresponding problem. The main question is: suppose we allow the network to operate for a very long time until it converges; can we do better than just getting some local optima? i.e., is it possible to design a network which will find the exact solution (or some guaranteed approximation) ?

In particular the following two questions are addressed:

1. Let L be an instance of an NP-hard problem; suppose that there exists a way to map L to a neural network N_L such that every local maximum of the energy function corresponds to a global optimum of the problem L. Clearly, the network will run for an exponential (in the input size) amount of time (worst case) until it will reach a stable state which corresponds to a solution of the problem.
   The question is: does there exist such a network N_L which has only polynomial (in the size of the input to L) number of neurons? The question is interesting because it is "known" how to build networks which will work very fast but are limited in the number of neurons.

2. Investigating the special case of the Traveling Salesman Problem (TSP) [3,5,6].

   Let E_{glo} be the energy value of the global maximum, and let E_{loc} be the energy value of a local maximum, we will say that the local maximum is an ϵ-approximate solution to the problem iff:

$$\frac{E_{glo} - E_{loc}}{E_{glo}} \leq \epsilon$$

   The question is: can we design a neural network which has a polynomial (in the size of the input) number of neurons in which the energy value of every local maximum is an ϵ-approximate of the solution to a given instance of TSP? That is, is it possible to design a neural network which will solve the TSP with some guaranteed approximation?

The main results in the paper are the answers to the above questions:

1. The answer to the first question is NO; more formally:

   Proposition 1 *The existence of a polynomial size neural network for solving an NP-hard problem will imply that NP = co-NP.*

2. The answer to the second question is NO; more formally:

   Proposition 2 *The existence of a polynomial size neural network for finding an ϵ-approximate solution to the TSP will imply that P=NP.*

The key observation for proving the above propositions is the fact that a single iteration takes time which is a polynomial in the number of neurons and in the size of the input to the corresponding problem. The proofs for the above two propositions rely on known results from complexity theory and the theory of local search algorithms (see [6] chapters 16,19).

References

[1] J. Bruck and J. Sanz, *A Study on Neural Networks*, IBM Technical Report, ARC Computer Science Department, RJ 5403, 1986.

[2] J. Bruck and J. W. Goodman, *A Generalized Convergence Theorem for Neural Networks and its Applications in Combinatorial Optimization*, IEEE First ICNN, San-Diego, June 1987.

[3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.

[4] J. J. Hopfield, *Neural Networks and Physical Systems with Emergent Collective Computational Abilities*, Proc. Nat. Acad. Sci. . USA, Vol. 79, pp. 2554-2558, (1982).

[5] J. J. Hopfield and D. W. Tank, *Neural Computations of Decisions in Optimization Problems*, Biol. Cybern. 52, pp. 141-152, (1985).

[6] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Inc., 1982.

[7] J. C. Picard and H. D. Ratliff, *Minimum Cuts and Related Problems*, Networks, Vol 5, pp. 357-370, (1974).

END

FILMED

FEB. 1988

DTIC