SESSION 3

PAPER 6

PANDEMONIUM: A PARADIGM FOR LEARNING

by

DR. O. G. SELFRIDGE

BIOGRAPHICAL NOTE

Oliver G. Selfridge was born in London 10 May 1926. He studied at the Massachusetts Institute of Technology from 1942-1945, returning post-graduately from 1946-1950. After 2 years at Signal Corps Laboratories at Fort Monmouth, New Jersey, he joined Lincoln Laboratories in Group 34, Communication Techniques, of which he is now Group Leader.

PANDEMONIUM: A PARADIGM FOR LEARNING

by

DR. O. G. SELFRIDGE

INTRODUCTION

WE are proposing here a model of a process which we claim can adaptively improve itself to handle certain pattern recognition problems which cannot be adequately specified in advance. Such problems are usual when trying to build a machine to imitate any one of a very large class of human data processing techniques. A speech typewriter is a good example of something that very many people have been trying unsuccessfully to build for some time.

We do not suggest that we have proposed a model which can learn to typewrite from merely hearing speech. Pandemonium does not, however, seem on paper to have the same kinds of inherent restrictions or inflexibility that many previous proposals have had. The basic motif behind our model is the notion of parallel processing. This is suggested on two grounds: first, it is often easier to handle data in a parallel manner, and, indeed, it is usually the more "natural" manner to handle it in; and, secondly, it is easier to modify an assembly of quasi-independent modules than a machine all of whose parts interact immediately and in a complex way.

We are not going to apologize for a frequent use of anthropomorphic or biomorphic terminology. They seem to be useful words to describe our notions.

What we are describing is a process, or, rather, a model of a process. We shall not describe all the reasons that led to its particular formulation, but we shall give some reasons for hoping that it does in fact possess the flexibility and adaptability that we ascribe to it.

THE PROBLEM ENVIRONMENT FOR LEARNING

Pandemonium is a model which we hope can learn to recognize patterns which have not been specified. We mean that in the following sense: we present to the model examples of patterns taken from some set of them, each time informing the model which pattern we had just presented. Then, after some time the model guesses correctly which pattern has just been presented before

we inform it. For a large class of pattern recognition ensembles there has never existed any adequate written or statable description of the distinctions between the patterns. The only requirement we can place on our model is that we want it to behave in the same way that men observably behave in. In an absolute sense this is a very unsatisfactory definition of any task, but it may be apparent that it is the way in which most tasks are defined for most men. Lucky is he whose job can be exactly specified in words without any ambiguity or necessary inferences. The example we shall illustrate in some detail is translating from manually keyed Morse code into, say, typewritten messages. Now it is true that when one learns Morse code one learns that a dash should be exactly three times the length of a dot and so on, but it turns out that this is really mostly irrelevant. What matters is only what the vast army of people who use Morse code and with whom one is going to have to communicate understand and practise when they use it. It turns out that this is nearly always very different from school book Morse.

In the same way the only adequate definition of the pattern of a spoken word, or one hand-written, must be in terms of the consensus of the people who are using it.

We use the term pattern recognition in a broad sense to include not only that data processing by which images are assigned to one or another pattern in some set of patterns, but also the processes by which the patterns and data processing are developed by the organism or machine; we generally call this latter "learning".

PANDEMONIUM, IDEALIZED AND PRACTICAL

We first construct an idealized pandemonium *(fig. 1)*. Each possible pattern of the set, represented by a demon in a box, computes his similarity with the image simultaneously on view to all of them, and gives an output depending monotonically on that similarity. The decision demon on top makes a choice of that pattern belonging to the demon whose output was the largest.*

---

* This is an exact correlate of a communications system wherein given a received message $M(T)$ and a number of possible transmitted messages $M_i(T)$, that M_i is chosen, that is, deemed to have been transmitted, which minimizes $\int |M(T)-M_i(T)|^2 dT$. (Such a procedure is optimum under certain conditions). This integral is, as it were, the square of a distance in a signal phase space - *fig.2* - and thus that transmitted message is selected that is most similar to the received one.

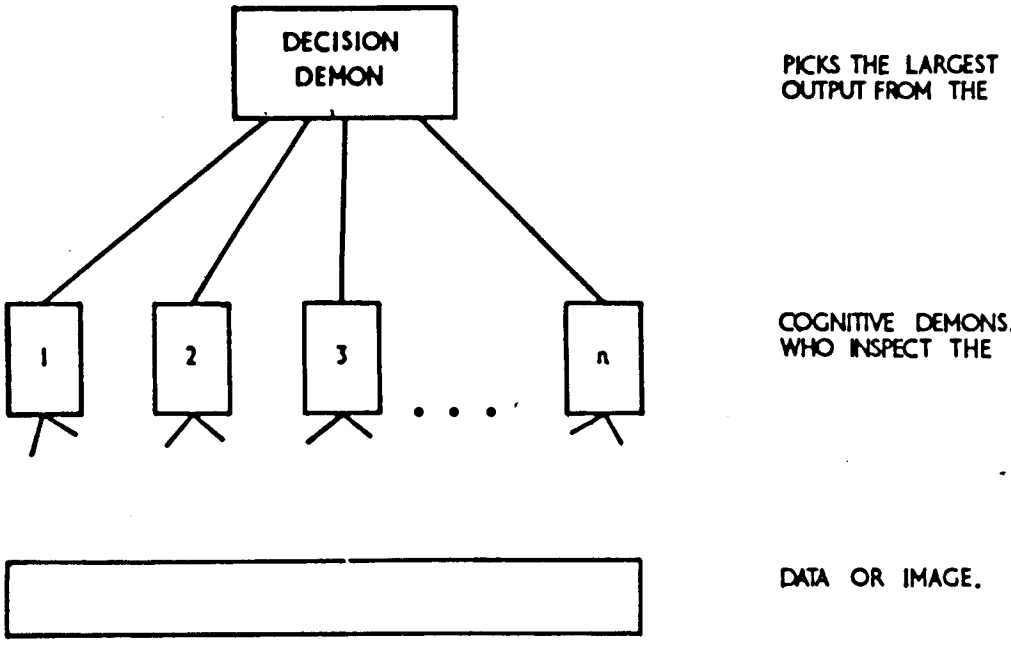

Fig. 1. Idealized Pandemonium.

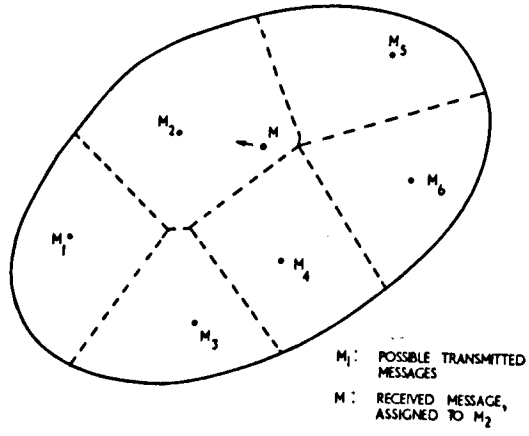

Fig. 2. Signal Space.

Each demon may, for example, be assigned one letter of the alphabet, so that the task of the A-demon is to shout as loud of the amount of 'A-ness' that he sees in the image.* Now it will usually happen that with a reasonable collection of categories - like the letters of the alphabet - the computations performed by each of these ideal cognitive demons will be largely the same. In many instances a pattern is nearly equivalent to some logical function of a set of features, each of which is individually common to perhaps several patterns and whose absence is also common to several other patterns.†

We therefor amend our idealized Pandemonium. The amended version - *fig. 3* - has some profound advantages, chief among which is its susceptibility to that kind of adaptive self-improvement that I call learning.

The difference between *fig. 1* and *fig. 3* is that the common parts of the computations that each cognitive demon carries out in *fig. 1* have in *fig. 3* been assigned instead to a host of subdemons. At this stage the organization has four levels. At the bottom the data demons serve merely to store and pass on the data. At the next level the computational demons or subdemons perform certain more or less complicated computations on the data and pass the results of these up to the next level, the cognitive demons who weigh the evidence, as it were. Each cognitive demon computes a shriek, and from all the shrieks the highest level demon of all, the decision demon, merely selects the loudest.

THE CONCEPTION OF PANDEMONIUM

We cannot ab initio know the ideal construction of our Pandemonium. We try to assure that it contains the seeds of self-improvement. Of the four levels in *fig. 3*, all but the third, the subdemons, which compute, are specified by the task. For the third level, therefore, we collect a large number of possible useful functions, eliminating a priori only those which could not conceivably be relevant, and make a reasonable selection of the others, being bound by economy and space. We then guess reasonable weights for them. The behaviour at this point may even be acceptably good, but usually it must be improved by means of the adaptive changes we are about to discuss.

---

* It is possible also to phrase it so that the A-demon is computing the distance in some phase of the image from some ideal A; it seems to me unnecessarily platonic to postulate the existence of 'ideal' representatives of patterns, and, indeed, there are often good reasons for not doing so.

† See, for example, Jerome Bruner, "A study of Thinking",

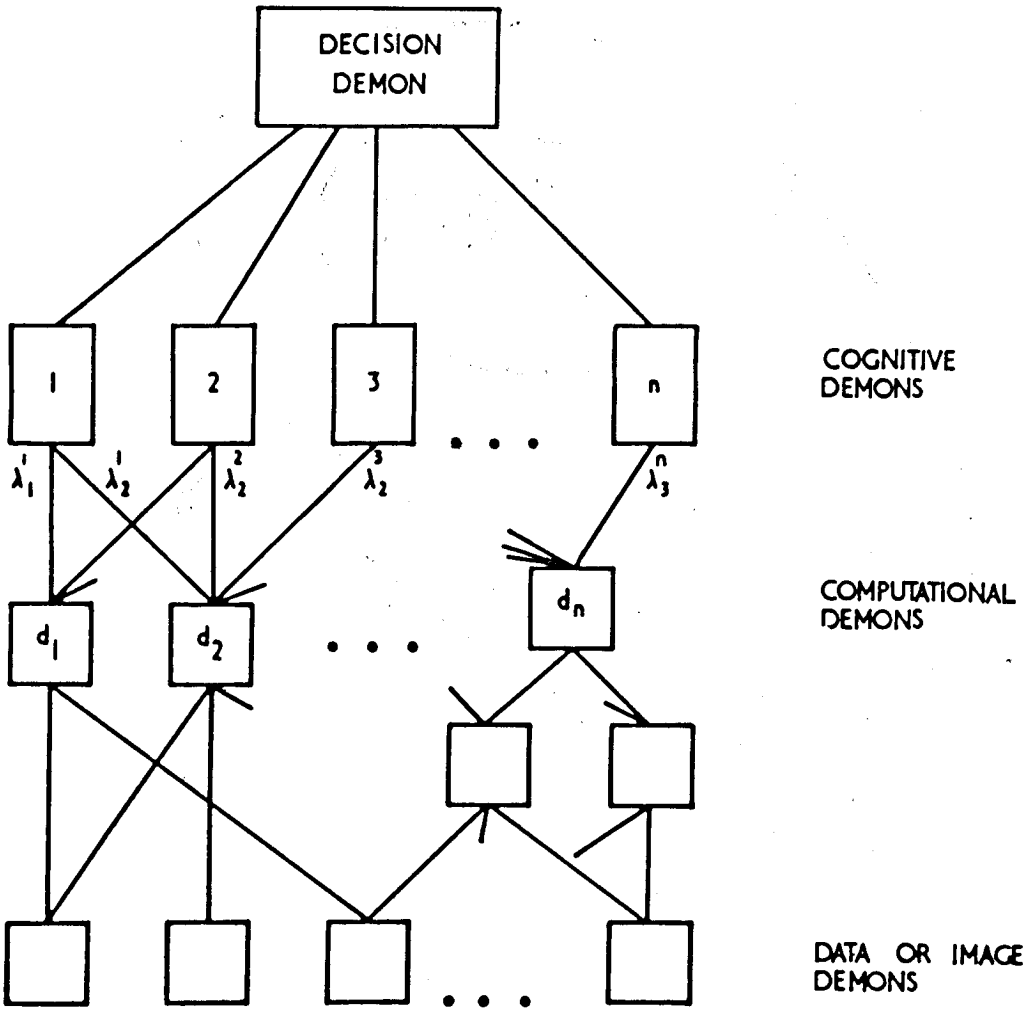

Fig.3. Amended Pandemonium

The Evolution of Pandemonium

There are several kinds of adaptive changes which we will discuss for our γ Pandemonium. They are all essentially very similar, but they may be programmed and discussed separately. The first may be called "Feature Weighting".

Although we have not yet specified what the cognitive demons compute, the sole task at present is to add a weighted sum of the outputs of all the computational demons or subdemons; the weightings will of course differ for

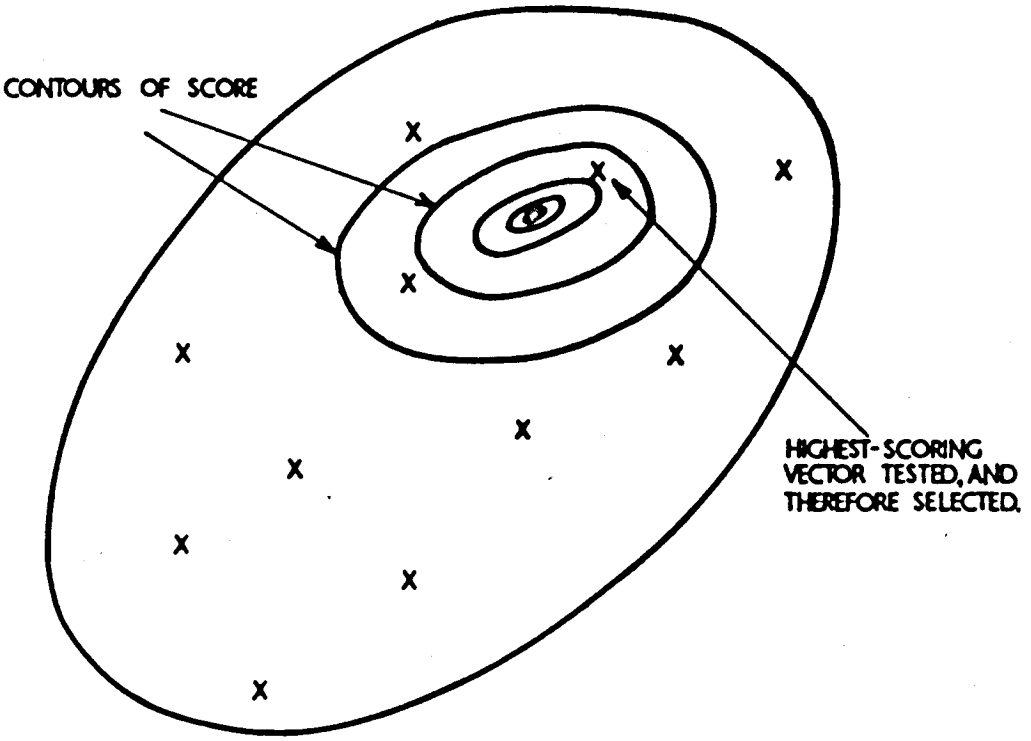

Fig.4. First hill climbing technique: pick vectors at random (points in the space), score them, and select the one that scores highest.

the different cognitive demons, but the weightings will be the only difference between them. Feature weighting consists of altering the weights assigned to the subdemons by the cognitive demons so as to maximize the score of the entire γ Pandemonium. How then can we do this?

The Score

What we mean by the score here is how well the machine is doing the task we want it to do. This presumes that we are monitoring the machine and telling it when it makes an error and so on, and for the rest of the discussion we shall be assuming that we have available some such running score. Now at some point we shall be very interested in having the machine run without that kind of direct supervision, and the question naturally arises whether the machine can meaningfully monitor its own performance. We answer that question tentatively yes, but delay discussing it till a later section.

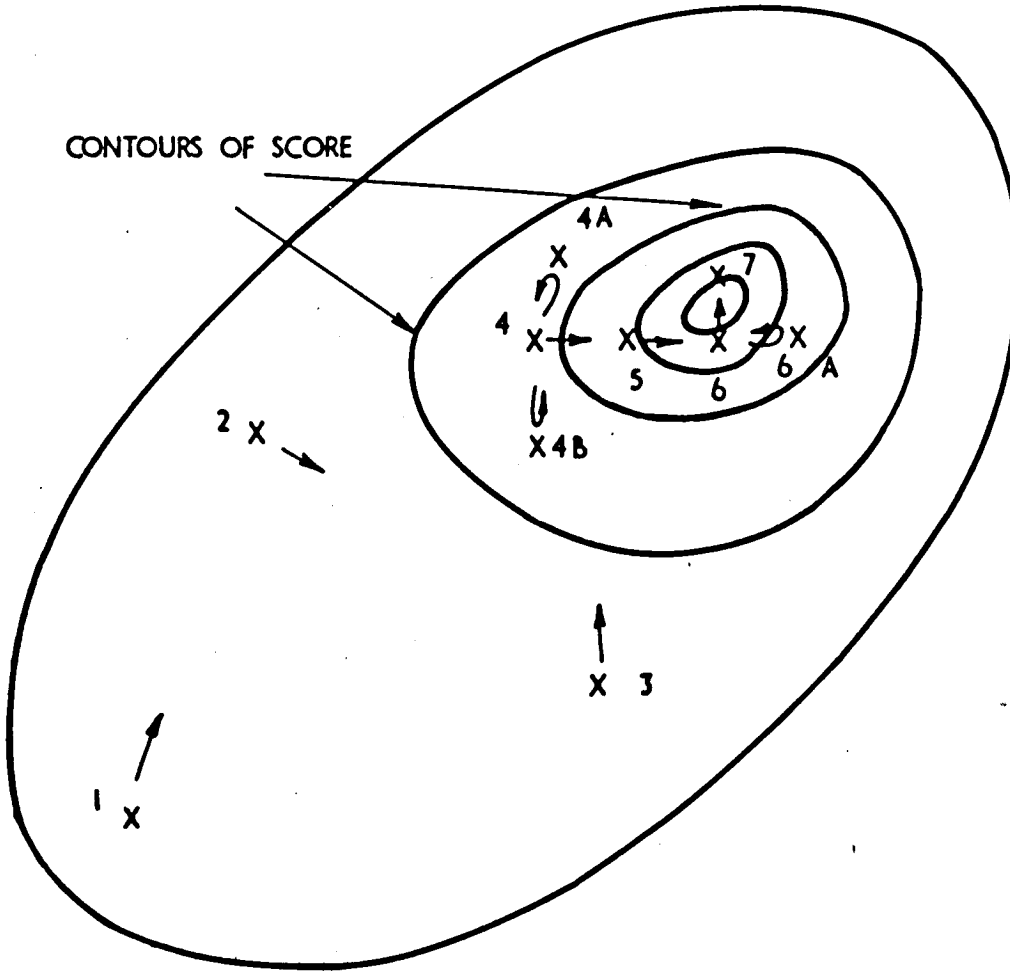

Fig.5. Second hill-climbing technique: pick vectors until one of them (Number 4) outscores the previous ones. Then take short random steps, retracing any that decrease the score.

Feature Weighting and Hill-Climbing

The output of any cognitive demon is

$$d_i = \Sigma \lambda^{i}_{j} d_j$$

so that the complete set of weights for all the cognitive demons over all the subdemons is a vector:

$$\Lambda = \left\{ \Lambda^i \right\} = \left\{ \lambda^1_1, \lambda^1_2 .. \lambda^2_1, \ldots \lambda^n_m \right\}$$

Now for some (unknown) set of weights Λ, the behaviour of this whole Pandemonium is optimum, and the problem of feature weighting is to find

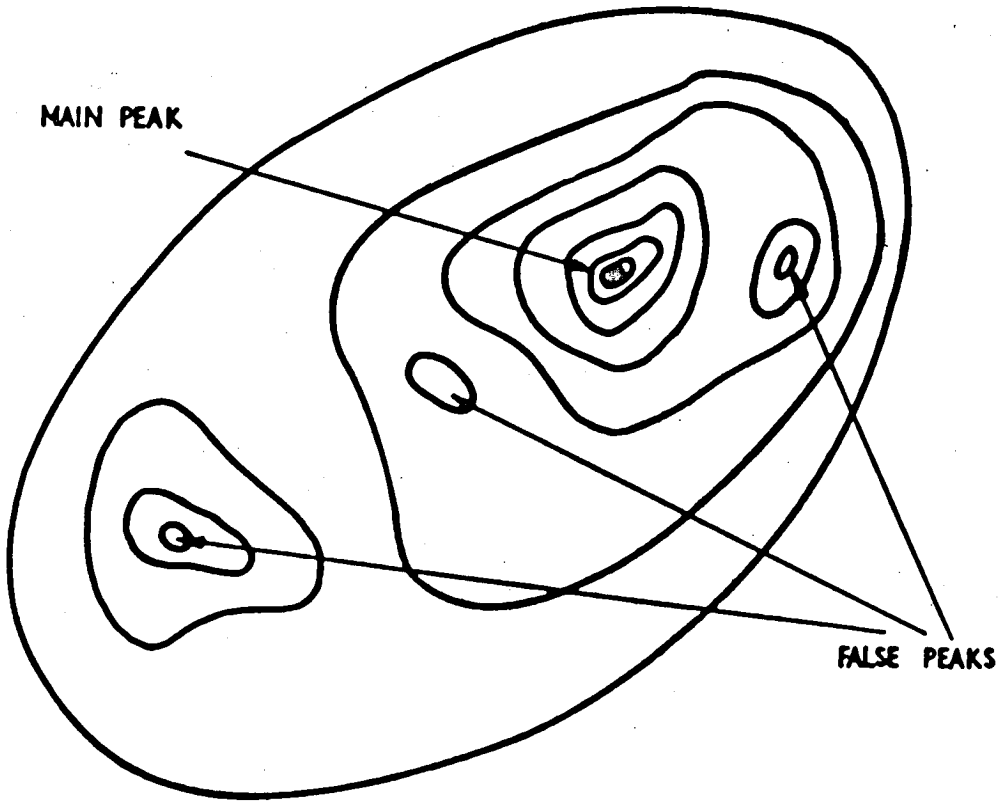

Fig.6. General space showing false peaks. One of the false peaks is quite isolated from the main or true peak.

that set. This may be described as a hill-climbing problem. We have a space (of Λ) and a function on the space (the score), which we may consider an altitude, and which we wish to maximize by a proper search through Λ. One possible technique is to select weighting vectors at random, score them, and finally to select the vector that scored highest (see *fig. 4*). It will usually, however, turn out to be profitable to take advantage of the continuity properties of the space, which usually exist in some sense, in the following way: select vectors at random until you find one that scores perceptibly more than the others. From this point take small random steps in all directions (that is, add small random vectors) until you find a direction that improves your score. When you find such a step, take it and repeat the process. This is illustrated in *fig. 5*, and is the case of a blind man trying to climb a hill. There may be, of course, many false peaks on which one may find oneself trapped in such a procedure (*fig. 6*).

The problem of false peaks in searching techniques is an old and familiar one. In general, one may hope that in spaces of very high dimensionality the interdependence of the components and the score is so great as to make very unlikely the existence of false peaks completely isolated from the main or true peak. It must be realized, however, that this is a purely experimental question that has to be answered separately for every hill-climbing situation. It does turn out in hill-climbing situations that the choice of starting point is often very important. The main peak may be very prominent, but unless it has wide-spread foot-hills it may take a very long time before we ever begin to gain altitude.

This may be described as one of the problems of training, namely, to encourage the machine or organism to get enough on the foot-hills so that small changes in his parameters will produce noticeable improvement in his altitude or score. One can describe learning situations where most of the difficulty of the task lies in finding any way of improving one's score, such as learning to ride a unicycle, where it takes longer to stay on for a second than it does to improve that one second to a minute; and others where it is easy to do a little well and very hard to do very well, such as learning to play chess. It is also true that often the main peak is a plateau rather than an isolated spike. That is to say, optimal behaviour of the mechanism, once reached, may be rather insensitive to the change of some of the parameters.

Subdemon Selection

The second kind of adaptive change that we wish to incorporate into our Pandemonium is subdemon selection. At the conception of our demoniac assembly we collected somewhat arbitrarily a large number of subdemons which we guessed would be useful and assigned them weights also arbitrarily. The first adaptive change, feature weighting, optimized these weights, but we have no assurance at all that the particular subdemons we selected are good ones. Subdemon selection generates new subdemons for trial and eliminates inefficient ones, that, is, ones that do not much help improve the score.

We propose to do this initially by two different techniques, which may be called "mutated fission" and "conjugation". The first point to note is that it is possible to assign a worth to each of the subdemons. It may be done in several ways, and we may, for example, write the worth W_i of the ith demon

$$W_i = \sum_j |\lambda_i^j|,$$

so that the worthy demons are those whose outputs are likely to affect most strongly the decisions made.

We assume that feature weighting has already run so long that the behaviour of the machine has been approximately optimized, and that scores and worths of machine and its demons have been obtained. First we eliminate those subdemons with low worths. Next we generate new subdemons by mutating the survivors and reweighting the assembly. At present we plan to pick one subdemon and alter some of his parameters more or less at random. This will usually require that we reduce the subdemon himself to some canonical form so that the random changes we insert do not have the effect of rewriting the program so that it will not run or so that it will enter a closed loop without any hope of getting out of it.*

Besides mutated fission, we are proposing another method of subdemon improvement called "conjugation". Our purpose here is two-fold: first to provide a logical variety in the functions computed by the subdemons, and, secondly, to provide length and complexity in them.

What we do is this: given two 'useful' subdemons, we generate a new subdemon which is the continuous analogue of one of the ten nontrivial binary two-variable functions on them. For example, the product of two subdemon outputs, corresponding to the logical product, would suggest the simultaneous presence of two features. The ten non-trivial such functions are listed in *Table 1*.

$A.B$	$A \vee B$
$A.\sim B$	$A \vee \sim B$
$\sim A.\sim B$	$\sim A \vee \sim B$
$\sim A.B$	$\sim A \vee \sim B$
$A.B \vee \sim A.\sim B$	$A.\sim B \vee \sim A.B$

Table 1. Non-trivial binary functions on two variables.

* We are at present running our Pandemonium on an IBM 704. The analogues for kinds of simulation are obvious.

Control Adaptation

The first two levels of adaptation are directly concerned with immediate improvement of behaviour and the score. We should also like to improve the entire organization, and in the same way. We shall deal with this point somewhat cursorily, being reluctant to specify things too far in advance of experiment. In principle, we propose that the control operationsshould themselves be handled by demons subject to changes like feature weighting and subdemon selection. It is obviously a little more difficult and perhaps impossible here to define the usefulness or worth of a particular demon. It is also clear that it will sometimes take much longer to check the usefulness of some change in some control demon - for example, in one of those which control the mutations in subdemon selection. Furthermore, at this level, some of the demons, presumably, will be in a position to change themselves, for otherwise we should need another level of possible change, and so on. This raises the possibility of irreversible changes, and it is not obvious that *all* parts of the machine should be subject to adaptive change. But these are largely heuristic questions.

The Evolutionary Process

The adaptive changes mentioned above will tend, we hope, to promote a kind of evolution in our Pandemonium. The scheme sketched is really a natural selection on the processing demons. If they serve a useful function they survive, and perhaps are even the source for other subdemons who are themselves judged on their merits.

It is perfectly reasonable to conceive of this taking place on a broader scale - and in fact it takes place almost inevitably. Therefore, instead of having but one Pandemonium we might have some crowd of them, all fairly similarly constructed, and employ natural selection on the crowd of them. Eliminate the relatively poor and encourage the rest to generate new machines in their own images.

Unsupervised Operation

So far all of the operation of the machine has been on the basis of constant monitoring by a person who is telling the machine when it makes an error. A very valid question is whether the machine can form any independent opinion of its own on how well it is doing. I suggest that it can in the following way: one criterion of correct decisions will be that they are fairly unequivocal, that is, that there is one and only one cognitive demon whose output far outshines the rest. Some running average of the degree to which this is so would presumably somewhat reflect the score of the machine. Note that it would be vital that the machine be trained first to do well enough before it is left to its own resources and supervision.

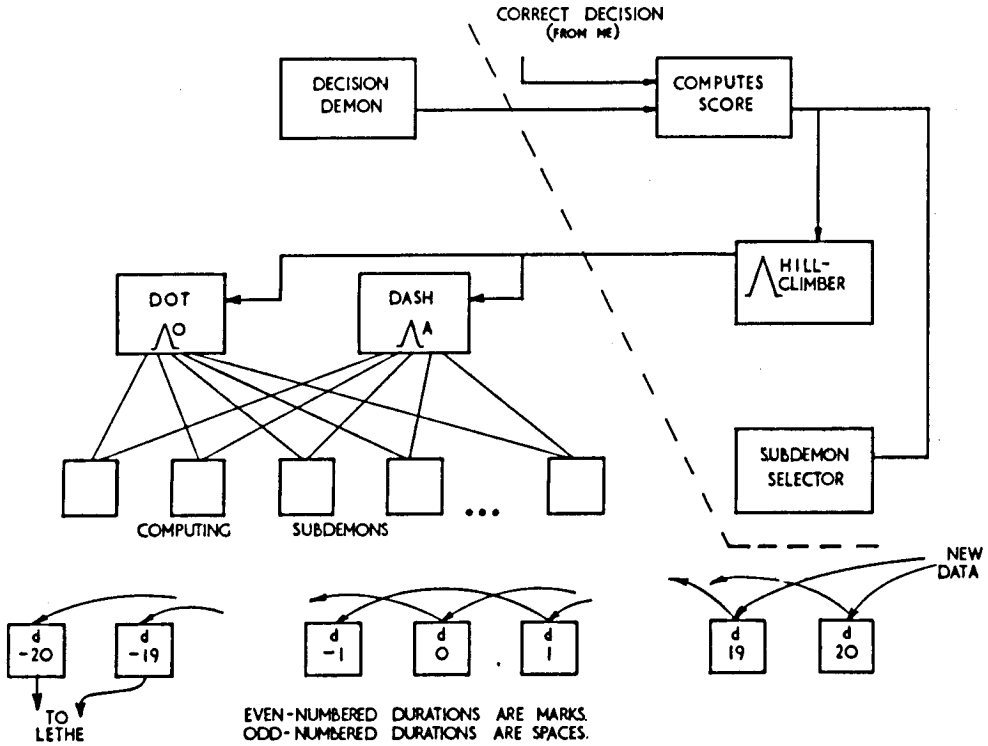

Fig.7. First morse pandemonium.

A REAL-LIFE EXAMPLE: MORSE TRANSLATION

As we mentioned before, the entire notion of Pandemonium was conceived as a practical way of automatically improving data-processing for pattern recognition. Our initial model task is to distinguish dots and dashes in manually keyed Morse code, so that our Pandemonium can be illustrated in *fig. 7*. Note that the functions and behaviour of all demons have been specified except for the computing subdemons. We shall reiterate those specifications.

(1) The decision demon's output is 'dot' or 'dash' according as the dot demon's output is greater or less than the dash demon's.

(2) The cognitive demons, dot and dash, each compute a weighted sum of the outputs of some 150 computing subdemons. Initial weights we have assigned arbitrarily, but, we hope, reasonably.

(3) The data-handling demons receive data in the form of durations, alternatively of marks and spaces, and they pass them down the line.

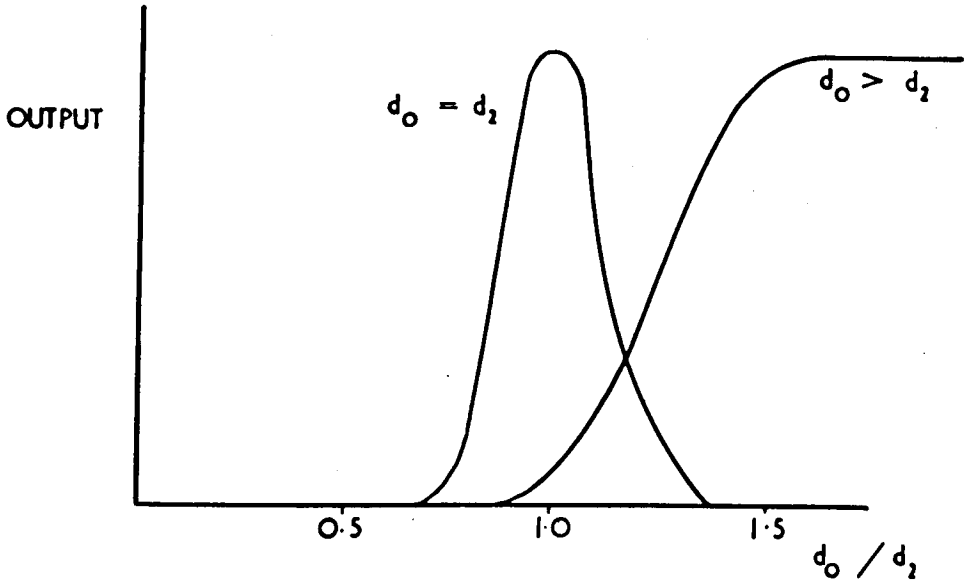

Fig.8. Operation of the subdemons $d_0 = d_2$, $d_0 > d_2$.

The computing subdemons are constructed from only a very few operational functions, which are carefully non-binary. For example the subdemons $d_0=d_2$ and $d_0>d_2$ have their outputs shown in *fig.8*. The operational functions follow:

(1) '='. This function computes the degree of equality of some set of variables (see *fig.8*).

(2) ' ', ' ', compute the degree to which some variable is less than or is greater than some other variable (see *fig.8*).

(3) '*max*', '*rmax*', compute the degree to which some variable is the largest of an arbitrary set of variables or an arbitrary set of consecutive variables.

(4) 'O_i', 'A_i', store the degree to which the ith duration *has been* identified as a dot or dash.

(5) '*Av*" computes an average of some set of variables.

(6) '*M*' is a family of tracking means. For example, it might compute

$$M^{\alpha}(c) = \alpha\, M(i-1) + (1-\alpha)$$

(7) '*Ox*', '*Ax*'. *Ox1* is the last duration identified as a dot. *Ax3* is the third last duration identified as a dash, etc.

The above is the present functional vocabulary of the computing operations for our subdemons. The subdemons themselves are built with a simple syntax. For the initial set, at conception, we merely select a set of operational functions and follow them with the numbers of some particular data demons.

CONCLUSION

What I shall present at the meeting in November will be the details of the progress of Pandemonium on the Morse code translation problem. The initial problem we have given the machine is to distinguish dots and dashes. When the behaviour of the machine has improved itself to the point where little further improvement seems to be occurring, we shall add three more cognitive demons, the symbol space, the letter space, and the word space. Presumably after some further time this new Pandemonium will settle down to some unimprovable state. Then we shall replace the senior or decision-making demon with a row of some forty or so character demons with a new decision-making demon above them, letting the new cognitive demons for the character demons use all the inferior demons, cognitive and otherwise, for their inputs. It is probably also desirable that previous decisions be available for present decisions, so that a couple of new functional operations might be added. There need be little concern about logical circularity, because we have no requirement for logical consistency, but are merely seeking agreeable Morse translation.

How much of the whole program will have been run and tested by November I cannot be sure of. At the present (July) we have had some fair testing of hill-climbing procedures.

ACKNOWLEDGEMENT

I should like to acknowledge the valuable contributions of discussions with many of my friends, including especially M. Minsky, U. Neisser, F. Frick and J. Lettvin.

DISCUSSION ON THE PAPER BY DR. O. G. SELFRIDGE

DR. J. W. BRAY: May I ask Dr. Selfridge what he thinks of this approach to his problem? Let x = duration of the last signal, which may be a dot or a dash, a signal space, letter space or word space. Let y = 2, if in fact it was a dash, 1 if dot, 0 if signal space, -1 if letter space and -2 if word space.
To form the polynomial:

$$y = A_0 + A_1x + A_2x^2 + A_3x^3 \ldots\ldots\ldots\ldots$$

take a number of observations, as he suggests, and let the machine learn the code by determining the coefficients A_0 etc. by simple curvilinear regression. The duration of previous and subsequent signals and the interpretation given to previous signals could be added as further variables on the right hand side.

MR. C. STRACHEY: At the end of the paper you promise us some further information about the latest state of the programme. Could you let us know what this is?

DR. J. McCARTHY: I would like to speak briefly about some of the advantages of the pandemonium model as an actual model of conscious behaviour. In observing a brain, one should make a distinction between that aspect of the behaviour which is available consciously, and those behaviours, no doubt equally important, but which proceed unconsciously. If one conceives of the brain as a pandemonium - a collection of demons - perhaps what is going on within the demons can be regarded as the unconscious part of thought, and what the demons are publicly shouting for each other to hear, as the conscious part of thought.

DR. D. M. MACKAY: Dr. Selfridge's 'pandemonium' has a certain family resemblance to a class of mechanism considered in some earlier papers *(ref. 1)* (though I had never suspected its demonic implications!).

In one of these, *(ref. 2)* after discussing the general principle that you penalise the unsuccessful, I pointed out that the amount of information

1. MACKAY, D. M. Mentality in Machines. *Proc. Arist. Soc. Suppt.*, 1952, 61.
2. MACKAY, D. M. The Epistemological Problem for Automata, ed. J. McCarthy and C. E. Shannan *Automata Studies, Princeton* (1955). See also *Brit. J. Psychol.*, 1956, **47**, 30 and *Advancement of Science*, 1956, 392.

per 'kick' (to use Dr. Selfridge's metaphor) is very small unless the probability of success and failure are equal. If you have a system where there are a vast number of possibilities to be eliminated by rather feckless trial and error, then of course failure occurs much more often than success. The solution I suggested was to form a kind of syndicated learning process in which at first large numbers of elements, destined eventually for independence, should be coupled together so as to reduce the diversity of response. A hand, for example, might not at first have each finger separately controllable, but could work clumsily as a whole. In that way, you can greatly decrease the amount of groping which is necessary before a successful adaptive action is found, although of course the degree of success achieved may be less. On this principle, if pursued to the limit, even the earliest trials could have a non-negligible chance of success, (though success in a very small way).

Given this easy start, simple self-organising subroutines can build up fairly quickly. As they increase their number and their success, however, the idea is that the couplings between elements should gradually be dissolved to increase the complexity of the problem. If you keep the complexity increasing step by step with the degree of development of successful internal matching sub-routines, then fully adaptive behaviour can be enormously more quickly developed than if the system starts with the full repertoire to be explored. My question is why Dr. Selfridge has not incorporated this principle in his 'pandemonium', so that each 'kick' could have something nearer to one bit of information instead of an almost negligible fraction.

DR. P. C. PRICE (written contribution): This is a very interesting and stimulating paper. I have one comment to make, and that is on the discussion of "Feature Weighting and Hill-Climbing".

I think that in this discussion the author has not brought out one important distinction between types of "hill-climbing" problems - that between determinate and stochastic problems. Whereas in a determinate problem the "hill" is defined by a single function of many variables

$$f(x) \equiv f(x_1, \ldots . x_n)$$

in a stochastic problem it is defined by the mean value of a number of functions:

$$f(x) \equiv f_\alpha(x)$$

when

$$f_\alpha(x) \equiv f(x_1, \ldots x_n;\ \alpha)$$

and α is a random variable whose value depends on the particular trial made.

Now I would have thought that most of the more interesting pattern recognition problems are essentially stochastic: the objective is to make

a machine that will obtain as high a proportion as possible of correct answers to a series of questions "what is this pattern?" referring to a random series of patterns.

The author has given an excellent account of some of the difficulties of determinate hill-climbing and of the techniques for overcoming them, but stochastic hill-climbing presents additional problems, and I think this may become very apparent when a pandemonium is built to deal with a practical task. In particular, two problems which will need investigation are:

(i) how large a sample of trial values $f_\alpha(x)$ and $f_\alpha(y)$, belonging to two points x and y in the vector space being explored, will be needed to obtain a satisfactory estimate of $[f(x) - f(y)]$ for the purposes of hill-climbing?

(ii) how much longer will a particular stochastic hill-climbing procedure take than the corresponding determinate procedure, and will this ratio prove to be too large, in some practical instances, to allow the evolutionary development of the pandemonium to take place?

I should be very interested to learn if the author has considered these problems and developed any solutions. The problem of stochastic hill-climbing is of practical interest outside the context of this paper, for instance in connection with the evolutionary operation of chemical processes.

DR. O. G. SELFRIDGE (in reply[ϕ]): Since I have been working on Morse code, which I am doing in addition to working on learning, I have met probably 50 people and when I say that I am working on a machine to do manually keyed Morse code, they say "I will tell you how to do that" and they then proceed to come up with some scheme. Actually, this is the first time I have heard this particular one, whereas I have come across the others many times, so Dr. Bray should be congratulated on a new scheme for solving Morse code. Let me assure him that it will not work; we have tried it. Morse code manually received is not that simple. The context necessary apparently extends at least 10 letters on *either* side - not just one side, both sides. In fact, the real question about doing this - why I chose Morse code - was exactly that it had had this kind of interactions with each other. You cannot do it by looking at binary functions of the durations, that is, expressing durations as binary digits and then looking on binary functions of lots of them. If I take 10 letters on either side, each has three marks, you should consider the spaces as well, but leaving those out, that is three times 20, which is 60 durations, and if we specify them to one part in 32, that is 300 bits, and binary functions of 300 bits cannot be picked at random. So the question is to take some steps in the right direction and then hope the steps are right enough so that you will get enough improvements, so that the following steps will get you even more so.

[ϕ]Edited versions by the author was not available.

I maintain that there is some merit in studying self-improvement systems and if I am going to do that I am going to study systems where I know what I want, rather than more difficult problems, however attractive they may be to mathematicians. Mathematicians like to work on unsolved problems for the greater glory, and problems already solved like the prime number theorem are left to graduate students. I have a suspicion that John McCarthy might later bring up some important points about descriptions, and here I see my point about solving useful problems, because Morse code is a useful problem. The whole question is at what level you are going to deal with descriptions of your data. In most problems, especially in dealing with the amount of things which people do, binary functions are just not adequate. For one thing there is too much data, and one of the first questions you talk about in the conditioned reflex - remember that conditional probability assumes that you have already recognised the stimulus; if you know that the stimulus belongs to one of a small class of functions, it is pretty easy, as a man, but mostly in human problems you do not know this.

Dr. Mackay made some very kind remarks. I would go further back than he did, as he well knows, in crediting demonology. All of us have sinned in Adam, we have eaten of the tree of the knowledge of good and evil, and the demonological allegory is a very old one, indeed. As for his remarks about the syndicate approach, the evolution of diversative organisation, I think that is an extremely accurate and good point. One of the things a species learns is not only how to survive but to have exactly the right variants in its members. The reason horseshoe crabs have not changed much is not because they don't adapt; they can adapt, but there is no variation, so they can only adapt very little, and they are about as good as they can be; whereas there are all kinds of people.

In speed and information theory, I really consider that speed in the classic sense of computation is so completely irrelevant to this problem - the number of binary operations one does a second - that it does not interest me very much though it interests computer designers. I would rather like to see lots of operations which could conceivably be done in parallel, being done sequentially, because this is the only machine we have, and then I hope people will have machines which will work in parallel, so that when I want a machine to do twice as difficult a problem I merely build twice as big a machine, instead of letting one machine work twice as long.

Dr. Strachey asked about results, and they are roughly as I have indicated. Improvement does take place; the hill-climbing does work. The sub-demon selection in the programmes that we have did not do what we hoped, but the sub-demons were effectively thrown out, and the ones which were kept were largely concentrated around those demons which relate those

ones very close. The most effective demon by itself, which worked 90 per cent. of the time, which was very surprising, was when a mark was the maximum, or within a just noticable difference of the maximum of six demons clustered round zero from minus 3 to plus 3. I am afraid that that is as much as one can say fairly.