PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT. © ICOT, 1988

ARTIFICIAL INTELLIGENCE AND NEURAL COMPUTING

Yuichiro Anzai
Department of Electrical Engineering, Keio University
3-14-1, Hiyoshi, Yokohama 223 Japan

ABSTRACT

The paper briefly discusses the relation between AI and parallelism, particularly from the viewpoint of distinction of three levels; interpretation, algorithm and implementation. It points out that, on one end, parallel symbolic AI models cover the levels of interpretation and algorithm, and, on the other end, the emerging paradigm for subsymbolic AI models with massively parallel computation encompasses the algorithm and implementation levels. The paper concludes that the effort for parallel AI should shed more light on the level of algorithm that is the place where the symbolic and subsymbolic approaches meet.

1 LEVELS FOR PARALLEL AI

Artificial Intelligence (AI) started with the belief that the human intelligence can be simulated on computing machines. Some of our beliefs were realized, but some are still behind the scene. Yet much effort has been invested for making the latter show up on stage. Parallel computation is regarded, for a long time, as an expected vehicle to bring them appear on the stage with universal applause.

Then, what are *they*? What kind of players are we expecting to be on stage? Do we just wish to make AI programs run much faster? Or do we have other Somethings in mind to run on future-day computers? The purpose of this paper is to briefly discuss these broad issues from a particular viewpoint, concluding that parallelists in AI should distinguish interpretation of intelligence and its implementation more clearly.

Towards this end, let me first introduce the level distinction for parallel AI. Distinction of levels has become popular, for example, by the work of Marr (1982) and Newell (1982). I believe that it is a key issue for clarifying the entangled relations between parallelism and AI. The following description of levels and their distinction into the *interpretation level*, *algorithm level*, and *implementation level*, does not precisely reflect the arguments of Marr, or Newell, but it may suffice for our issues.

First, the interpretation level refers to models of AI that are drawn by our mind, directly reflecting our conscious interpretation of intelligence. For example, models of grammatical rules for natural language (Chomsky, 1965) are at the interpretation level. State-space models of problem solving (Newell and Simon, 1972) are also at the interpretation level. In this vein, Poggio and his colleagues' interpretation of early vision as regularization of ill-posed problems (Poggio and Koch, 1985) is a formulation of vision process at the interpretation level, though they call it the level of computational theory.

Second, the algorithm level refers to the formulation of interpretation-level models by computational algorithms. For instance, the same set of grammatical rules can be computationalized for syntactic analysis by a serial top-down DCG algorithm (Pereira and Wallen, 1980), or a parallel bottom-up PAX algorithm (Matsumoto, 1986). Also, very different parallel or serial algorithms for tree search can represent the same problem solving models at the algorithm level.

Third, the implementation level refers to the executable organization of computational algorithms on software and hardware architectures. For example, a parallel marker propagation algorithm, which is a formulation at the algorithm level, can be executed on a specific kind of semantic network machines at the implementation level (Fahlman, 1979). The introduction of shared and read-only variables in concurrent logic programming (Shapiro, 1983) also corresponds to the implementation level.

2 LEVEL DISTINCTION

The distinction of three levels introduces qualitatively different possibilities for parallel AI. In this section , I briefly discuss pros and cons of those possibilities. Just to simplify discussion, I would like to classify the current approaches to AI into five directions: (1) logic, (2) heuristic search, (3) constraint satisfaction, (4) memory-based reasoning, and (5) neural modeling. Also, because of the character of this paper (served for a panel), I only summarize my personal view below, pointing only to directly relevant references. Readers interested in more of specific results may consult other references, e.g., Fahlman (1988).

2.1 Interpretation Level

First, note that conventional AI work with logic and heuristic search is mostly concerned not with parallel, but with serial models at the level of interpretation. The reason for this is that most of them model human thought, which is inherently serial at the level of consciousness. Even most of the work under the banner of distributed problem solving employs serial models, when we see them at the interpretation level.

On the other hand, virtually all of AI models based on constraint satisfaction, memory-based reasoning, and neural modeling more or less incorporate parallelism. However, most of those models are concerned not with conscious, but with rapid, subliminal information processing such as perception and memory retrieval.

The survey of research in AI suggests that parallelization of AI models for thinking and other higher-level human information processing at the interpretation level has not attained sufficient success. It is interesting to see that those models are represented typically in symbolic terms.

2.2 Algorithm Level

The algorithm level is essentially independent from the interpretation level. Different serial or parallel algorithms can represent the same interpretation-level model. However, we see that, at least in some areas of research, the former draws heavily on the characteristics of the latter.

Particularly, it is a common phenomenon that the seriality of models for human thought is brought down to the algorithm level. Working memories in production systems (Forgy *et al.*, 1984), blackboards in blackboard architectures (Nii, 1986), and global control of hypotheses in belief revision systems (de Kleer, 1986) are examples for this. Also, the parallelism of models for rapid subconscious processing is typically inherited to the algorithm level. Examples include relaxation labeling in image understanding (Waltz, 1975), data-parallel algorithms in memory-based reasoning (Stanfill and Waltz, 1986), and stochastic relaxation methods for image recovery (Geman and Geman, 1984).

One of the important issues in AI and parallelism is how to evaluate the algorithm level, particularly to what degree it should be independent from the interpretation level. More concretely, the problem is to what extent we should reflect our own interpretation of intelligence directly on the structure and semantics of algorithms.

2.3 Implementation Level

The implementation level is also independent from the algorithm level. The same algorithm can be implemented on radically different software or hardware architectures. However, there emerge two opposite camps to leave apart from there.

One camp believes that, to obtain a sufficient speed for computation, the semantic gap of the two levels should be diminished as much as possible. Examples are the designs of special-purpose neural chips and image-processing machines. The other camp accepts it with credit that architectures can be just independent from algorithms, and it is enough to design general-purpose machines that can execute any algorithms fast. Examples are recent advances of massively-parallel computers with processors of various granularity levels. Another example is parallel implementation of a variety of symbol processing languages, from Lisp and Prolog to production systems and object-oriented languages.

From the viewpoint of traditional symbolic AI, the implementation level is the most remote from the serious consideration, since it puts primal emphasis on modeling at the interpretation and algorithm levels, and deemphasizes their connections to the level of implementation. The emerging new paradigm of massively parallel computing algorithms and their implementation techniques have started to break such belief down at least partially. The intelligence is not occupied entirely by internal processing of a given limited number of symbols. The core of intelligence lies in its dynamic interaction with the outside world. Thus, the implementation level, which must support such interaction, should be directly related not only with the algorithm level, but also with the interpretation level.

2.4 Interlevel Relations

We have seen that, though the three levels, interpretation, algorithm and implementation, are essentially independent, we need to consider models that encompass more than one level to realize more intelligent machines. An obstacle against it is that most of the models for symbolic intelligence are not brought down to the implementation level, and most for rapid, unconscious intelligence do not climb up to the interpretation level.

One good sign for rescuing this gap is that both approaches cover the level of algorithms. Especially, if algorithms from both sides share the same semantics, it will open up the possibility of integrating both directions. In the following section, I describe some examples suggesting this good symptom.

3 EXAMPLES

Here, I provide three examples, all from the work on neural computing, that might suggest possible integration of levels for parallel AI.

3.1 Modular Neural Networks

One of the simplest tasks that call for both of higher-level reasoning and lower-level perception is series extrapolation problem solving. For example, to find a letter that occupies _ in the alphabetical series, *abmcdmefm_*, cooperation of serial reasoning to discover the underlying abstract representation, and rapid parallel processing to find patterns like $m...m...m$ helps to hypothesize the regularity as quickly as possible. In other words, the latter works, at the interpretation level, as the *generator of hypotheses* for periodicity of a given series. Thus, if algorithms for the former and the latter match in semantics, then it provides a good example of the integration of the interpretation and implementation levels grounded at the level of algorithms.

In this regard, we have devised a computational model of the former, reasoning process, applying a neural network with local representation (Anzai *et al.*, 1987). Its algorithm is based on spreading activation and micro features (Waltz and Pollack, 1985). Also, we have constructed a model of the latter, perceptual process, using neural modules with distributed representation (Anzai and Shimada, 1988). The model consists of a position-independent shape-recognizing module, shape-independent position-recognizing module, and a module for integrating them. The three modules work based on extensions of von der Malsburg's competitive learning algorithm (von der Malsburg, 1973).

The coupling of those two models is made by activating nodes of the reasoning model by the outputs of the perception model. The integrated model is modularized, and organized by sparse interconnections, at the level of algorithms.

3.2 Discourse Processing

Waltz and Pollack (1985) pioneered in applying neural networks to resolving ambiguity of natural language meanings. We are taking similar direction, with some modifications, in Japanese discourse processing (Tamura and Anzai, 1987). These pieces of work cover the interpretation and algorithm levels, and make some suggestions for a relation to the implementation level.

Ambiguity resolution in our system is made by the mixed computation of rule-based symbolic processing and neural-network-based spreading activation. The former is used for constructing *part-of-speech instance networks*, and the latter is applied to resolving ambiguity. The representation and control of networks are made in the environment of OPHELIA, an object-oriented knowledge representation language that can represent hierarchical sets of production systems (Anzai and Kondo, 1986).

The system reads each word w from the top of a given sentence, and executes the following steps in order: (1) generates an instance of w, (2) activates the *concept class* of w, and generates and activates a *concept instance* of w, (3) activates the *feature class* of w, and generates and activates a *feature instance* of w, (4) generates and activates a *part-of-speech instance* of w, (5) spans a part-of-speech instance network, and (6) performs spreading activation.

The steps (1)-(6) are executed for each of the consecutive words in the given sentence. The meaning of the sentence is defined as the pattern of activation levels distributed over the nodes in the network given at the time when the steps for the last word are completed. Japanese syntax is represented by production rules that are included in one of OPHELIA objects. These rules are used in step (5) to span networks for part-of-speech instances.

We have applied our system to resolution of Japanese ambiguity in simple discourse contexts, where the meaning of the second sentence is determined by that of the first.

Systems of the kind mentioned above are based on the interpretation-level model that natural language discourse ambiguity can be resolved by the cooperation of symbolic and associative processes. At the algorithm level, those processes are represented by execution of rules, and activation of nodes, respectively. At the implementation level, each processing unit corresponds to an object, and message passing procedures implement the exchange of link weights between those objects. Thus, this example from resolution of Japanese ambiguity exhibits a model that covers all of the three distinct levels.

3.3 Problem Solving by Neural Networks

One of popular applications of neural networks lies in finding suboptimal solutions of complex discrete optimization problems. In these applications, a problem is formulated typically as the minimization of some energy function, which is radically different from symbolic models for problem solving. This new approach even results in an easy implementation of the model as analog circuits (Hopfield, 1984). Though this approach, including its followers like Boltzmann Machines (Ackley, Hinton and Sejnowski, 1985) and Cauchy Machines (Szu, 1987), does not simulate the conscious process of human problem solving, it is able to cover both the algorithm and implementation levels.

Thus, as the last example, I present a particular piece of work on Packing Puzzle problem solving by neural networks (Akiyama *et al.*, 1988). Packing Puzzle consists of an $m \times n$ board, and tiles of various shapes. The goal of the problem is to exactly pack the board by the tiles without overlapping.

The model is a neural network with three parameters: *reference activation level* a_0, temperature T, and discrete time step Δt. Those parameters are related by the following four equations (Akiyama and Yamashita, 1988):

$$
\begin{aligned}
net_i &= \sum_{j=1}^{N} w_{ij} o_j + \theta_i + \varepsilon, \\
\sigma &= kT, \\
\frac{\Delta a_i}{\Delta t} &= -\frac{a_i}{\tau} + net_i, \\
o_i &= f(a_i) = \frac{1}{2}(tanh(\frac{a_i}{a_0}) + 1),
\end{aligned}
$$

where net_i is the total input to neuron i, w_{ij} is the link weight from neuron j to i, θ_i is the input bias for neuron i, ε is the Gaussian noise with the average 0 and variance σ^2, $k = \sqrt{8/\pi}$, a_i is the activation level of neuron i, τ is the time constant, and f is the output function.

The model, called Gaussian Machines, can thus be represented as $GM(a_0,\ T,\ \Delta t)$. Gaussian Machines can be regarded as a generalization of McCulloch-Pitts model ($GM(0,0,1)$), Hopfield model ($GM(a_0,\ 0,\ \Delta t)$), and Boltzmann Machines ($GM(0,T,1)$). Gaussian Machines have succeeded in solving Packing Puzzles with boards with reasonable sizes, using simulated annealing for modifying T, and *sharpening* for adjusting a_0.

Furthermore, Gaussian Machines have been implemented on specially fabricated neural chips, realizing variable conductance on Hopfield circuit by deterministic and stochastic switched resistor circuits (Akiyama, 1988).

Gaussian Machines touch upon at least two levels of interpretation, algorithm and implementation. Whether the model is rigorously defined at the interpretation level depends on whether we interpret the behavior of Gaussian Machines for solving the puzzle as intelligent or not.

3.4 Summary

Summarizing the analysis in this section, we observe that all the three distinct levels are more or less taken into account by the examples from modular networks, and discourse processing. But neural network models for problem solving seem to lack the interpretation level.

Note that the former two examples include both of symbolic and subsymbolic processes. Thus, it is natural that they incorporate all the three levels. However, it should be noted that such integration could be made by using the algorithm level formulation. Also, note that the third example for Packing Puzzle introduced only the subsymbolic representation. It caused the model not directly related to the interpretation level. To make it related, it may need be tied with some symbolic model at the program level.

4 OTHER ISSUES

There exist many important problems, other than the level distinction issue, for parallelism and AI. Examples include:

1. Knowledge representation: how does a parallel AI model divide the labor between control and data. This control-parallel and data-parallel issue is deeply related to the design of MIMD and SIMD AI architectures.
2. Granularity of processors: which is better for parallel AI, small-grained or large-grained processors? Does the choice depend heavily on problem domains?
3. Interprocess communication: which is better for parallel AI, sparse or dense interconnections of processors?
4. Interprocess structure: which is better for parallel AI, flat, hierarchical, or any other structure of processor allocations?
5. Semantic gaps: Which is better for parallel AI, general-purpose or special-purpose hardware architectures?

Most of the issues left untouched here, including those listed above, are on parallelism in AI-oriented languages and hardware architectures. It is interesting to observe that most of good results on symbolic parallel AI are restricted to designs of languages and architectures, and relatively few reported on *models*. I feel that this observation is not accidental, but caused by an inherent nature of human intelligence.

5 CONCLUDING REMARKS

Three levels for parallelism in AI were discussed, and the importance of the intermediate, algorithm level for integrating serial/symbolic and parallel/subsymbolic models of AI was emphasized with some examples.

I have not discussed directly issues related to AI languages and architectures. It is because these issues, though very important to realize fast computing machines, are not inherent in AI research. Rather, one of the most essential problems in parallelism and AI at present is how to organize the interpretation and implementation levels at the algorithm level in a naturalistic way.

Again, I should remark that the view presented here points out, based on presently available data, only one particular issue, just among many. Forecasting the future of parallel AI is not a trivial task, since research areas related to it are diverged but strongly connected. Its nontriviality may even be comparable with pinpointing the weather of the day one year after. But parallel computation is crucially different from weather: the former can be developed by our own effort. The future for parallel AI can be designed by ourselves.

References

[1] Ackley,D.H., Hinton,G.E. and Sejnowski,T.J.: A learning algorithm for Boltzmann Machines, *Cognitive Science*, **9**, 147-169, 1985.

[2] Akiyama,Y.: Gaussian Machines neural model and its analog and digital architectures, *IEICE Report*, CPSY88-16, 1988.

[3] Akiyama,Y. and Yamashita,A.: Gaussian Machines: A general neuron model, *Proc. of SICE'88*, August, 1988.

[4] Akiyama,Y., Yamashita,A. and Kajiura,M.: Application of Gaussian Machines to optimization problems: Examples from packing puzzle, Paper presented at the 10th Neural Information Science Meeting, August, 1988.

[5] Anzai,Y. and Kondo,T.: Object-oriented production-embedded knowledge representation language OPHELIA and its applications, *Computer Software*, **3**, 43-60, 1986.

[6] Anzai,Y., Mori,H., Ito,M. and Hayashi,Y.: A serial-parallel integrated information-processing model for complex human problem solving, In G.Salvendy (ed.), *Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems*, Elsevier Science Publishers, 1987.

[7] Anzai,Y. and Shimada,T.: Modular neural networks for shape and/or location recognition, *Proc. of the 1st Annual Conference of INNS*, September, 1988.

[8] Chomsky,N.: *Aspects of the Theory of Syntax*, The MIT Press, 1965.

[9] de Kleer,J.: An assumption-based TMS, *Artificial Intelligence*, **28**, 127-162, 1987.

[10] Fahlman,S.E.: *NETL: A System for Representing and Using Real-World Knowledge*, The MIT Press, 1979.

[11] Fahlman,S.E.: Parallel processing in artificial intelligence, In J.S.Kowalik (ed.), *Parallel Computation and Computers for Artificial Intelligence*, Kluwer Academic Publishers, 1988.

[12] Forgy,C.L, Gupta,A., Newell,A. and Wedig,R.: Initial assessment of architectures for production systems, *Proc. of AAAI-84*, 1984.

[13] Geman,S. and Geman,D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. on Pattern Anal. and Machine Intell.*, **PAMI-6**, 721-741, 1984.

[14] Hopfield,J.J.: Neurons with graded response have collective computational properties like those of two-state neurons, *Biophysics, Proc. Natl. Acad. Sci. USA*, **81**, 3088-3092, 1984.

[15] Marr,D.: *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H.Freeman, 1982.

[16] Matsumoto,Y.: A parallel parsing system for natural language analysis, *Proc. of the 3rd International Conference on Logic Programming*, London, 1986, *Lecture Notes in Computer Science*, **225**, Springer, 396-406, 1986.

[17] Newell,A.: The knowledge level, *Artificial Intelligence*, **18**, 87-127, 1982.

[18] Newell,A. and Simon,H.A.: *Human Problem Solving*, Prentice-Hall, 1972.

[19] Nii,H.P.: Blackboard systems, and blackboard application systems from a knowledge engineering perspective, *AI Magazine*, August, 1986.

[20] Pereira,F.C.N. and Warren,D.H.D.: Definite clause grammars for language analysis: A survey of formalism and a comparison with augmented transition networks, *Artificial Intelligence*, **13**, 231-278, 1980.

[21] Poggio,T. and Koch,C.: Ill-posed problems in early vision: from computational theory to analogue networks, *Proc. of Royal Society of London*, **B226**, 303-323, 1985.

[22] Shapiro,E.: A subset of Concurrent Prolog and its interpreter, *ICOT Technical Report*, TR-003, 1983.

[23] Stanfill,C. and Waltz,D.: Toward memory-based reasoning, *Comm. ACM*, **29**, 1213-1228, 1986.

[24] Szu,H.: Fast simulated annealing, *Physics Letters A*, **122**, 157-162, 1987.

[25] Waltz,D.: Generating semantic descriptions from drawings of scenes with shadows, In P.H.Winston (ed.), *The Psychology of Computer Vision*, McGraw-Hill, 1975.

[26] Waltz,D. and Pollack,J.B.: Massively parallel parsing: A strongly interactive model of natural language interpretation, *Cognitive Science*, **9**, 51-74, 1985.

[27] Tamura,A. and Anzai,Y.: Natural language processing system based on connectionist models, *J. of Information Processing Society of Japan*, **28**, 202-210, 1987.

[28] von der Malsburg, C.: Self-organization of orientation sensitive cells in the striate cortex, *Kybernetik*, **14**, 85-100, 1973.