Massively Parallel Artificial Intelligence

Hiroaki Kitano (Chairperson)
Carnegie Mellon University NEC Corporation
Pittsburgh, PA, 15213 Tokyo 108, Japan
hiroaki@cs.cmu.edu

James Hendler
University of Maryland, USA
hendler@cs.umd.edu

Tetsuya Higuchi
Electrotechnical Laboratory, Japan
higuchi@etl.go.jp

Dan Moldovan
University of Southern California, USA
moldovan@gringo.usc.edu

David Waltz
Thinking Machines Corporation, USA and
Brandeis University, USA
waltz@think.com

Abstract

Massively Parallel Artificial Intelligence is a new and growing area of AI research, enabled by the emergence of massively parallel machines. It is a new paradigm in AI research. A high degree of parallelism not only affects computing performance, but also triggers drastic change in the approach toward building intelligent systems; memory-based reasoning and parallel marker-passing are examples of new and redefined approaches. These new approaches, fostered by massively parallel machines, offer a golden opportunity for AI in challenging the vastness and irregularities of real-world data that are encountered when a system accesses and processes Very Large Data Bases and Knowledge Bases. This article describes the current status of massively parallel artificial intelligence research and positions of each panelist.

1 Introduction

The goal of the panel is to highlight current accomplishments and future issues in the use of massively parallel machines for artificial intelligence research, a field generally called *massively parallel artificial intelligence*. The importance of massively parallel artificial intelligence has been recognized in recent years due to three major reasons:

1. increasing availability of massively parallel machines,
2. increasing interest in memory-based reasoning and other highly-parallel AI approaches,
3. development efforts on Very Large Knowledge Bases (VLKB).

Despite wide recognition of massively parallel computing as an important aspect of high performance computing and general interest in the AI community on highly parallel processing, only a small amount of attention has been paid to exploring the full potential of the massive parallelism offered on currently available machines. One of the causes of this is that little communication has occurred between hardware architects and AI researchers. Hardware architects design without actually recognizing the processing, memory, and performance requirements of AI algorithms. AI researchers have developed their theories and models assuming idealizations of massive parallelism. Further, with few exceptions, the AI community has often taken parallelism as a mere "implementation detail" and has not yet come up with algorithms and applications which take full advantage of the massively parallelism available.

The panel intends to rectify this situation by inviting panelists knowledgeable and experienced in both hardware and application aspects of massively parallel computing in artificial intelligence. There are two interrelated issues which will be addressed by the panel: (1) the design of massively parallel hardware for artificial intelligence, and (2) the potential applications, algorithms and paradigms aimed at fully exploring the power of massively parallel computers for symbolic AI.

2 Current Research in Massively Parallel AI

2.1 Massively Parallel Machines

Currently, there are a few research projects involving the development of the massively parallel machines and a few commercially available machines being used for symbolic AI. Three projects of particular importance are:

- The CM-2 Connection Machine (Thinking Machines Corporation),
- The Semantic Network Array Processor (University of Southern California)
- The IXM2 Associative Memory Processor (Electrotechnical Laboratory, Japan).

These machines provide an extremely high-level of parallelism (8K - 256K) and promise even more in the future. Table 1 shows the specification of these machines.

Machine	CM-2	SNAP-1	IXM2
Architecture	Array Processor SIMD	Array Processor MIMD/SIMD	Associative Processor MIMD/SIMD
Connection	Hyper-Cube	Modified Hyper-Cube	Full Connection
PE Type	1 bit PE	TMS320C30 DSP	T800 Transputer
PE Number	64K	160	64
Maximum Parallelism	64K	160	256K
Maximum Nodes	64K (VPR = 1)	16K	256K
Word Length	1 bit	32 bit	32 bit
Peak Throughput	28 GFlops (single precision)	3.4 GFlops	7.2 GOPS
Communication Channels	24576	96 8-bit channels	224 Serial links
Channel Bandwidth	5 Mbit/sec	32 MBits/sec	19.2 MBits/sec
Broadcast/Collection Bus	1 bit	32 bit	32 bit and 2.4Mbyte/sec serial
Application Controller	Vax/SUN	Custom Dual Processor	SUN
Message Processing	Foreground	Background	Foreground
Message Control	Central Controller	Local via Propagation Rules	Local

Table 1: Comparison of the major massively parallel machines

There are two major approaches to designing a massively parallel machine: the Array Processor and the Associative Processor. CM-2 and SNAP are examples of the array processor architecture, and IXM2 is an example of the associative processor architecture. While the array processor architecture attains parallelism by the number of physical processors available, the associative processor attains parallelism by the associative memory assigned to each processor. Thus, the parallelism attained by the associative processor architecture is beyond the number of processors in the machine, whereas the array processor attains parallelism equal to the number of actual processors. This is why the IXM2 attains 256K parallelism with 64 processors. However, operations carried out by associative memories are limited to bit-marker passing and relatively simple arithmetic operations. When more complex operations are necessary, the parallelism will be equal to the number of processors.

Regarding the parallelism, the next version of the IXM2 (may be called IXM3) will aim at over one million parallelism using up-to-data processors and high density associative memory chips. The SNAP project is planning to develop a custom VLSI to attain a one million processor-scale machine. DARPA (Defense Advanced Research Projects Agency) is funding a project to attain TeraOps by 1995 [Waltz, 1990].

2.2 Massively Parallel AI Paradigm

In addition to designing new hardware architectures, the strategies and perhaps even the paradigms for designing and building AI systems may need to be changed in order to take advantage of the full potential of massively parallel machines. Emergence of massively parallel machines offers a new opportunity for AI in that large-scale DB/KB processing can be made possible in real-time. Waltz's talk at AAAI-90 [Waltz, 1990] envisioned the challenge of massively parallel AI. Two of the major ideas that play central roles in massively parallel AI are: memory-based reasoning and marker-passing.

Memory-based reasoning and case-based reasoning assume that memory is a foundation of intelligence. Systems based on this approach store a large number of memory instances of past cases, and modify them to provide solutions to new problems. Computationally, the memory-based reasoning is an attractive approach to AI on massively parallel machines due to the memory-intensive and data-parallel nature of its operation. Traditional AI work has been largely constrained by the performance characteristics of serial machines. Thus, for example, the memory efficiency and optimization of serial rule application has been regarded as a central issue in expert systems design. However, massively parallel machines may take away such constraints by the use of highly parallel operations based on the idea of data-parallelism. The memory-based reasoning fits perfectly with this idea.

Another approach is marker-passing. In the Marker-Passing approach, knowledge is stored in semantic networks, and objects called *markers* are propagated, in parallel, to perform the inference. Marker-passing is a powerful method of performing inferencing on large semantic network knowledge-bases on massively parallel machines, due to the high degree of parallelism that can be attained. One obvious application of this approach is the processing of Very Large Knowledge Bases (VLKB) such as MCC's CYC [Lenat and Guha, 1989], EDR's electric dictionaries [EDR, 1988] (both of which are expected to require millions of network links), and ATR's dialogue database [Ehara et. al., 1990]. It is clear that as KBs grow substantially large (over a million concepts) the complex (and often complete) searches used in many traditional inferencing systems will have to give way to heuristic solutions unless a high degree of parallelism can be exploited. Thus, the use of massively parallel computers for VLKB processing is clearly warranted.

Table 2 shows massively parallel AI systems developed so far. The list is by no means exhaustive, only lists the major systems. Also, there are many other models which match well with massively parallel machines. But, we only list the systems that are actually implemented on massively parallel machines.

System	Task	Machine	Reference
CIS	Concurrent Inference System	CM-1	[Blelloch, 1986]
CFS	Classifier System	CM-1	[Robertson, 1987]
MBRtalk	Word pronunciation by MBR	CM-1	[Stanfill, 1988]
Parallel ATMS	Massively Parallel ATMS	CM-1	[Dixon and de Kleer, 1988]
PHI-PSI	Protein Structure Prediction by MBR	CM-2	[Zhang, et. al., 1988]
PARADYME	Parallel Case Retrieval for CBR	CM-2	[Kolodner, 1988]
DowQuest	Information Retrieval	CM-2	[Stanfill et. al., 1989]
PARKA	Frame-based AI language	CM-2	[Evett et. al., 1990a]
Census	Census data classification by MBR	CM-2	[Creecy et. al., 1990]
PRA*	Massively Parallel Heuristic Search	CM-2	[Evett et. al., 1990b]
ASTRAL	Machine Translation (a version of ΦDMDIALOG)	IXM2	[Kitano and Higuchi, 1991a]
DmSNAP	Machine Translation (a version of ΦDMDIALOG)	SNAP	[Kitano, et. al., 1991b]
GA-1	Rule Learning by Genetic Algorithms	IXM2	[Kitano, et. al., 1991a]

Table 2: Massively Parallel AI Systems

3 Associative Memory Architecture

Tetsuya Higuchi
Electrotechnical Laboratory, Japan

In this talk, we consider the architectural requirements for massively parallel AI applications, based on our experiences of developing a parallel associative processor IXM2 and applications for IXM2. In addition, we introduce the current status of the Electric Dictionary Research project in Japan which is a real example of a very large knowledge base containing 400,000 concepts.

We have developed a parallel associative processor IXM2 which enables 256K parallel operations using a large associative memory. IXM2 consists of 64 associative processors with 256K word large associative memory and 9 communications processors. These are interconnected based on a complete connection scheme to improve marker propagations. Due to its bit-parallel nature, the associative memory is more powerful in fundamental operations of AI such as association and set intersection, compared with 1-bit PEs of SIMD machines like Connection Machine [Thinking Machine Corp., 1989], MPP [Batcher, 1980] and DAP [Bowler, 1984].

The current applications for IXM2 include:
(1) very large knowledge base processing,
(2) memory-based parsing for real-time speech-to-speech translation, and
(3) rule-based learning system using genetic algorithms.

As we develop applications for IXM2, we also compare the results on IXM2 with those on other high performance machines such as Connection Machine (CM-2), Cray-XMP and SUN-4 in order to investigate the architectural requirements for massively parallel AI applications.

Now we enumerate some findings through our experiments.

1. Supercomputers are not necessarily fast for applications of knowledge base processing and memory-based parsing.

   Example 1. Set Intersection: Set intersection can be performed in O(1) on SIMD machines like IXM2 and CM-2, because the data-level parallelism can be utilized by direct mapping of datum to each processing element. On the other hand, supercomputers perform it in O(N). Therefore, there is a difference of two orders of magnitude in execu- tion time between Cray-XMP and CM-2 for 64K data, and a difference of three orders between Cray-XMP and IXM2.

   Example 2. Marker Propagation: Marker propagation is intensively used in processing is-a hierarchy know- ledge base. It actually traverses links of the network structured data. A marker propagation program written in C, which uses recursive procedure call for traversing links, was run both on SUN-4 and Cray-XMP. In spite of the exactly same program, Cray was slightly slower than SUN-4. The main reasons for this are that the overhead of recursive procedure calls is heavy, and that network structured data can not be represented well with array data structures which best fit Cray.

2. Performances on SIMD are heavily influenced by the number of simultaneous activations of communications.

   SIMD machines prefer applications where:
   (1) Computation can be done in parallel on each PE, and
   (2) Communications between PEs are local and the communication can be done in parallel.

   This is because SIMD machines employ 1-bit PEs and serial(slow) communication links between PEs. Applications with above characteristics are often found in scientific computations. However, AI applications are not necessarily the case. AI applications where all PEs are not always active and the number of simultaneous communications are a few often cause the severe degradation in performance. According to our experiments on knowledge base processing and memory-based parsing, CM-2 is the best for applications with "average" simultaneous communication over 1,000. And for applications under 1,000, IXM2 outperforms CM-2. However, it seems that AI applications with simultaneous activations over 1,000 are not commonly seen.

3. Interaction overheads between the host and SIMD machines.

   The rule-based learning system using genetic algorithms (classifier systems) is one of the typical examples which require frequent interactions between the host processor and SIMD machine. In such applications, the per-

formances of SIMD machines are degraded heavily by the interaction overheads; the communication bandwidth and efficiency between the host and parallel processing modules have to be designed carefully to alleviate the problem. In addition, the introduction of the processing capability located in an intermediate level between the host and parallel processing modules may be very effective for this problem as demonstrated in dedicated architectures for image processing. Medium-grain multicomputers operating in MIMD mode, such as IXM2, MIT J-machine [Dally et. al., 1989], and iWarp [Borkar et. al., 1990] are also promising in this respect.

Inference algorithms to VLKB have to be investigated and evaluated using large-scale knowledge bases such as CYC and EDR. Practical knowledge bases include many exceptions (cancellation of inheritance) and tangled is-a hierarchy. Without such examples, it is very hard to develop efficient and robust inference algorithms.

The EDR electric dictionaries are the promising environment where investigations for VLKB processing techniques should be conducted. The dictionaries consist of a word dictionary, concept dictionary, co-occurrence dictionary and bilingual dictionary. The concept dictionary is especially interesting to VLKB researchers. It contains knowledge on the 400,000 concepts defined by the word dictionary. The knowledge is described in a form similar to semantic network.

4 How to Design a Marker-Passing Architecture for Knowledge Processing

Dan Moldovan
Computer Engineering
University of Southern California
Los Angeles, CA 90089-0781

In this talk we will share our experience in designing a parallel marker-passing computer system dedicated for processing semantic network applications. Over the last few years we have investigated and eventually implemented such a system. It is called SNAP (Semantic Network Array Processor).

We have approached this problem by first understanding the processing requirements of some AI domains and then seeking computer structures to satisfy these requirements. The outcome of our design effort was a parallel computer architecture capable of performing marker and value passing. Some of the architectural innovations of the SNAP machine are its unique high-level instruction set, marker propagation rules, and processor architecture.

A SNAP prototype has been implemented in our laboratory using off the shelf components. The prototype has 160 microprocessors grouped into some 32 clusters. It is capable of storing 16 k node semantic network with approximately 160 k inter-node relations.

The primary application for the SNAP machine is Natural Language Processing. We have found out that SNAP is suitable for NLP. In particular there is a good match between SNAP's distributed memory with its marker-passing features and the new dynamic memory parsing approach. For some limited domains we have observed parsing speeds in the order of millisecond per sentence [Kitano, et. al., 1991b].

5 Massively Parallel AI Applications[1]

David L. Waltz[2]
Thinking Machines Corporation and
Brandeis University

Memory-based and Case-based reasoning methods fit perfectly on massively parallel computers of all varieties; these methods use analogies with previous examples to decide on appropriate courses of action for new examples. In order for memory-based methods to work, one needs, in general, a database of previous examples, along with a "shell" that contains the machinery for matching new with previous examples. The database is of exactly the same sort that is typically used to train artificial neural nets or AI learning systems such as ID3.

Such systems have now been applied to a number of real-world applications; an MBR system that automatic classifies US Census Bureau returns will be described. This system significantly outperforms an expert system devised for the same task, but is most noteworthy because the effort to build it was only about 1/50th that required to build the expert system. Other MBR systems show promise for handling problems that have generally been considered to require rule-based solutions; for instance, Sumita and Iida have recently demonstrated the value of MBR-like methods for machine translation [Sumita and Iida, 1991].

I will argue that for nearly every domain of AI interest, MBR is likely to be more appropriate than rule-based methods. This is because most domains contain both regularities (that seem to encourage rule-based approaches) as well as large number of exceptions or idiosyncrasies (that demand item-by-item treatment). Unfortunately for those who favor rules, the ubiquity and sheer number of exceptions may cause the number of rules needed to handle all phenomena to become extremely large, so large that the number of rules is on the same order as the number of phenomena. MBR systems handle both regularities and exceptions in a uniform and simple-to-program fashion. Trade-offs between different learning and knowledge engineering methods will be discussed, along with implications of new and more powerful hardware and other factors.

6 Massively Parallel Symbolic AI[3]

James A. Hendler
University of Maryland

It has been argued that memory-based reasoning can best be performed on a parallel platform by the application of an associative-memory-type process running over a database of training examples. It is my contention that while such approaches may be useful in applications, they fall far short of

[1]This research was funded in part by the Defense Advanced Research Projects Agency, administered by the U.S. Air Force Office of Scientific Research under contract number F49620-0058, and in part by the United States Bureau of the Census.

[2]Thinking Machines Corporation, 245 First St., Cambridge, MA, 02142, USA.

[3]This research was funded in part by the ONR grant N-00014-88-K-0560 and NSF grant IRI-8907890. Development of the PARKA project has been performed in conjunction with two of my students, Matthew Evett and Lee Spector.

the inferencing needs of complex AI systems. If we are truly to succeed at NLP, planning, and other tasks requiring a richness of knowledge, we will have to automate the sorts of complex inferencing procedures that have been the mainspring of work in the traditional AI symbolic reasoning paradigm. Although the majority of the AI research done to date on actual parallel platforms has focused on vision research or on connectionist modeling, I will demonstrate that symbolic inferencing, in the form of traditional AI frame systems, can also show significant performance gains when using massive parallelism.

My discussion will center on a frame-based knowledge representation system, called PARKA, which runs on the massively parallel Connection Machine. Our research to date has centered on demonstrating that PARKA's performance of common types of inferencing can be far superior to that of serial systems. We have concentrated on two types of inferencing, bottom-up and top-down inheritance, both related to ISA-hierarchy property inheritance. Property inheritance is at the heart of most representation systems. Designing PARKA to have superior performance on property inheritance calculations furnishes a solid platform on which to base PARKA's other representation mechanisms.

For "top-down" inheritance queries, those which must start at the root of the tree and proceed towards the leaves (for example, "what are all the animals") we see that PARKA has worst-case runtime of $O(d)$; linear with respect to the depth d of the network, while serial inheritance programs have a worst-case performance of $O(B^d)$ (where B is the average branching factor in the network). For relatively large networks (over 32K nodes and upwards of 100K links) PARKA can process top-down inheritance queries in under two seconds.

We are currently working on extending the representational power of the PARKA language. One important ability which we are now focusing on is the ability to perform recognition queries, which we will argue are necessary to performing case-based inferencing with any real generality. We will describe a method by which the PARKA system can handle complex recognition queries in time approximating $O(D + M)$, where M is the number of conjuncts in the query. This contrasts dramatically with the $O(M \times B^d)$ time taken by current systems. I will argue that such algorithms are necessary to the success of large "common sense" knowledge-bases, such as the US CYC project or the Japanese electronic dictionary.

7 Designing Massively Parallel AI Systems[4]

Hiroaki Kitano
Carnegie Mellon University and
NEC Corporation, Japan

This talk addresses some of the issues that the designer of the massively parallel AI systems should notice. Some of the issues affect design decisions of the overall design ideas and some issues affects choice of the hardware.

1. Gaining massive parallelism
2. Deciding where to gain parallelism
3. Mapping from logical world to physical world
4. Avoiding PE overload
5. Minimizing Communication

First, a high level of parallelism needs to be attained in order to take advantage of the massively parallel machines. However, if we simply map current AI systems which extensively rely on piecewise rule applications, the level of parallelism attained can only be medium at best. The memory-based approach fits perfectly with massively parallel machines because matching of an input against all cases will be considered in parallel by SIMD operation. For example, the traditional view of natural language processing has been relying upon grammar rules to analyze sentences. However, in reality, natural language is a bulk collection of exceptions, and many serious NLP systems have a large set of rules which cope with each exceptions. Memory-based parsing and memory-based translation theory is a superior and practical model for building practical NLP systems to be delivered to the real-world.

Second, the designer should notice that not all processes can be parallelized. The processes which can be parallelized differ from one architecture to another. For example, the array processor architecture (such as CM-2 and SNAP) can parallelize activation of more than one nodes, but thise architecture does not send markers in parallel – each marker will be send out sequentially from one node. The associative processor (such as IXM2) can send markers in parallel, but activation of nodes will be in serial in each driving PE, thus parallelism will be only 64 for this operation. In some cases, creating/deletion of nodes and links requires controller interrupts which makes this part of process serial. Designers should be well aware of the characteristics of each architecture and should avoid turning a massively parallel machine into a serial machine.

Third, logical structure of the semantic networks is not necessary mapped directly on physical allocation. Suppose we have a node with 10 fanout. All 10 neighbour nodes are within one hop on the logical map. However, if the PE has only 4 physically connected neighbour PEs, at least 6 of the logical neighbours will be allocated on PEs which are more than one hop.

Fourth, there are hardware constraints. For example, if marker-passing algorithm requires propagation of addresses or other information, each node needs to have memory to store the information, or the information will be simply lost. Physical constraints on the memory capacity limits numbers of markers which can be legally acceptable to each node. In addition, fine-grained massively parallel machines do not have powerful PEs assigned to each node, so that heavy operations such as unification would kill entire performance of the system.

Fifth, minimization of communication is critical in designing high performance massively parallel AI systems. Although the massively parallel machine circumvented a Von-Neumann bottleneck, it encounters a communication bottleneck. In some case, over 95% of entire computing time was consumed in communication of data between processors. Physically, a communication between processor is an expensive operation.

These are some of the design issues for massively parallel AI applications. This list may give the impression that designing a massively parallel AI system is a hard task, but it is not true. It simply requires a paradigm change of the view toward intelligent processes. We had been relying on

[4]This research was funded by the National Science Foundation under grant MIP-9009111, and by the Pittsburgh Supercomputing Center under grant IRI-910002P.

somewhat rule-based and serial thinking, which may be due to hardware constraints of the serial machines we have so far. The alternative view which is more oriented toward memory-based and parallel thinking, enables us to build more practical AI applications, and once one get used to massively parallel thinking, and it would be a viable alternative to many of the current AI approaches.

References

[Batcher, 1980] Batcher, K., "Design of a Massively Parallel Processor," *IEEE Transactions on Computers* 29, no.9, September 1980.

[Blelloch, 1986] Blelloch, G. E., *AFS-1: A programming language for massively concurrent computers*, Tech Report 918, MIT AI Lab., 1986.

[Blelloch, 1986] Blelloch, G. E., "CIS: A Massively Concurrent Rule-Based System," *Proceedings of the National Conference on Artificial Intelligence (AAAI-86)*, 1986.

[Borkar et. al., 1990] Borkar, S., Cohn, R., Cox, G., Gross, T., Kung, H. T., Lam, M., Levine, M., Moore, B., Moore, W., Peterson, C., Susman, J., Sutton, J., Urbanski, J., and Webb, J., "Supporting Systolic and Memory Communication in iWarp", *The 17th Annual International Symposium on Computer Architecture*, May 1990.

[Bowler, 1984] K.C.Bowler and G.S.Pawley, "Molecular Dynamics and Monte Carlo Simulation in Solid-State and Elementary Particle Physics," *Proceedings of the IEEE*, 74, January 1984.

[Creecy et. al., 1990] Creecy, R., Masand, B., Smith, S., and Waltz, D., "Trading MIPS and Memory for Knowledge Engineering: Automatic Classification of Census Returns on a Massively Parallel Supercomputer," Thinking Machine Corp., 1990.

[Dally et. al., 1989] Dally, W. J., et. al. "The J-Machine: a fine-grain concurrent computer," Information Processing 89, Elsevier North Holland, 1989.

[Dixon and de Kleer, 1988] Dixon, M. and de Kleer, J., "Massively Parallel Assumption-based Truth Maintenance," *Proceedings of the the National Conference on Artificial Intelligence (AAAI-88)*, 1988.

[Drumheller, 1986] Drumheller, M., "Connection Machine stereo-matching," *Proceedings of the National Conference on Artificial Intelligence (AAAI-86)*, 1986.

[EDR, 1988] Japan Electric Dictionary Research Institute, "EDR Electric Dictionaries," Technical Report, Japan Electric Dictionary Research Institute, 1988.

[Ehara et. al., 1990] Ehara, T., Inoue, N., Kohyama, H., Hasegawa, T., Shohyama, F., and Morimoto, T., *Content of the ATR Dialogue Database*, TR-I-0186, ATR Interpreting Telephony Research Laboratories, 1990.

[Evett et. al., 1990a] Evett, M., Hendler, J., and Spector, L., *PARKA: Parallel knowledge representation on the Connection Machine*, Tech Report CS-TR-2409, University of Maryland, 1990.

[Evett et. al., 1990b] Evett, M., Hendler, J., Mahanti, A., and Nau, D., "PRA* A memory-limited heuristic search procedure for the connection machine," *Proceedings – Frontiers of Massively Parallel Computing*, 1990.

[Higuchi et. al., 1991] Higuchi, T., Kitano, H., Furuya, T., Kusumoto, H., Handa, K., and Kokubu, A., "IXM2: A Parallel Associate Processor for Knowledge Processing," *Proceedings of the National Conference on Artificial Intelligence*, 1991.

[Hillis, 1985] Hillis, D., *The Connection Machine*, The M.I.T. Press, 1985.

[Inmos, 1987] Inmos, *IMS T800 Transputer*, 1987.

[Kitano, et. al., 1991a] Kitano, H., Smith, S. F., and Higuchi, T., "GA-1: A Parallel Associative Memory Processor for Rule Learning with Genetic Algorithms," *Proceedings of the International Conference on Genetic Algorithms (ICGA-91)*, 1991.

[Kitano, et. al., 1991b] Kitano, H., Moldovan, D., Um, I., Cha, S., "High Performance Natural Language Processing on Semantic Network Array Processor," *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI-91)*, 1991.

[Kitano and Higuchi, 1991a] Kitano, H. and Higuchi, T., "High Performance Memory-Based Translation on IXM2 Massively Parallel Associative Memory Processor," *Proceeding of the National Conference on Artificial Intelligence (AAAI-91)*, 1991.

[Kitano and Higuchi, 1991b] Kitano, H. and Higuchi, T., "Massively Parallel Memory-Based Parsing," *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI-91)*, 1991.

[Kitano, 1990] Kitano, H., "ΦDMDIALOG: A Speech-to-Speech Dialogue Translation," *Machine Translation*, 5, 301-338, 1990.

[Kitano, 1988] Kitano, H., "Multilinguial Information Retrieval Mechanism using VLSI," *Proceedings of RIAO-88*, Boston, 1988.

[Kolodner, 1988] Kolodner, J., "Retrieving Events from a Case Memory: A Parallel Implementation," *Proceedings of Case-Based Reasoning Workshop*, 1988.

[Lenat and Guha, 1989] Lenat, D. B. and Guha, R. V., *Building Large Knowledge-Based Systems*, Addison-Wesley, 1989.

[Moldovan et. al., 1990] Moldovan, D., Lee, W., and Lin, C., *SNAP: A Marker-Passing Architecture for Knowledge Processing*, Technical Report PKPL 90-4, Department of Electrical Engineering Systems, University of Southern California, 1990.

[Riesbeck and Martin, 1986] Riesbeck, C. and Martin, C., "Direct Memory Access Parsing," *Experience, Memory, and Reasoning*, Lawrence Erlbaum Associates, 1986.

[Robertson, 1987] Robertson, G., "Parallel Implementation of Genetic Algorithms in a Classifier System," Davis, L., (Ed.) *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann Publishers, 1987.

[Stanfill et. al., 1989] Stanfill, C., Thau, R., and Waltz, D., "A Parallel Indexed Algorithm for Information Retrieval," *Proceedings of the SIG-IR-89*, 1989.

[Stanfill and Waltz, 1988] Stanfill, C. and Waltz, D., "The Memory-Based Reasoning Paradigm," *Proceedings of the Case-Based Reasoning Workshop*, DARPA, 1988.

[Stanfill and Waltz, 1986] Stanfill, C. and Waltz, D., "Toward Memory-Based Reasoning," *Communications of the ACM*, 1986.

[Stanfill, 1988] Stanfill, C., "Memory-Based Reasoning Reasoning Applied to English Pronunciation," *Proceedings of the National Conference on Artificial Intelligence (AAAI-88)*, 1988.

[Sumita and Iida, 1991] Sumita, E., and Iida, H., "Experiments and Prospects of Example-Based Machine Translation," *Proceedings of the Annual Meeting of the ACL (ACL-91)*, 1991.

[Thinking Machine Corp., 1989] Thinking Machine Corporation, *Model CM-2 Technical Summary*, Technical Report TR-89-1, 1989.

[Waltz, 1987] Waltz, D., "Application of the Connection Machine," *Computer*, Jan., 1987.

[Waltz, 1990] Waltz, D., "Massively Parallel AI," *Proceedings of the National Conference on Artificial Intelligence (AAAI-90)*, 1990.

[Waltz and Stanfill, 1988] Waltz, D. and Stanfill, C., "Artificial Intelligence Related Research on the Connection Machine," *Proceedings of the International Conference on Fifth Generation Computer*, 1988.

[Zhang, et. al., 1988] Zhang, X., Waltz, D., and Mesirov, J., *Protein Structure Prediction by Memory-based Reasoning*, Thinking Machine Corporation, 1988.

Some articles relevant to the topic, but not referred in the text, are also included for the reader's convenience in further investigation of the subject.