Viewpoint

Herbert A. Simon and Toshinori Munakata

AI Lessons

In commenting on the recent chess match between Garry Kasparov and Deep Blue, IBM has avoided speaking of AI, focusing almost exclusively on its large parallel processor that examines about 1.8×10^{10} positions in each three minutes (the average time allowed for a move). IBM promotional literature even proclaimed, "The power behind Deep Blue is an IBM RS/6000 SP system finely tuned with customized processor chips designed by IBM Research. This combination, in addition to expert knowledge, enables users to take on larger problems by analyzing a greater number of possible solutions."

We have a somewhat different view of Deep Blue's prowess and its implications for computing in general and AI in particular. Grandmaster Joel Benjamin, a consultant to the Deep Blue team, and Murray Campbell, an IBM research scientist and computer-chess expert, claimed important advances during the year between matches with Kasparov in Deep Blue 's chess knowledge, not merely its ability to examine more positions.

Deep Blue's branching factor at each move (or ply) averages perhaps four or five, after allowing for various procedures, like the alpha-beta algorithm and special continuations that are selectively incorporated into the search. Hence, although the speed of Deep Blue was increased by a factor of two during the year between matches, the search capability increased by only a fraction of a ply. Yet all grandmasters watching the match (as well as Kasparov) commented on the *qualitative* change in its play, especially its ability to play differently in different kinds of positions. So it is to chess knowledge, not brute-force search, we must look for most of the improvement. That knowledge appears to take three forms:

- A large opening "book," probably containing tens of thousands of game trees from actual games and from Deep Blue's own analyses, comparable to the amount of specific information about openings possessed by Kasparov and other grandmasters, although its evaluations of book positions are probably less exact than Kasparov's.
- The ability to assign valuations (also only approximate but subject to improvement) to each of the leaf positions it reaches through look-ahead search.
- Perhaps most important, its ability to notice patterns of pieces distinguishing one kind of position from another and to use different weights for features in evaluating positions of different character (so, for example, a bishop may be worth more with one arrangement of pawns than with another). It also seems capable of adjusting the selectivity and perhaps direction of its search to the character of the position; under certain circumstances, it conducts very deep but narrow selective searches.

Thus there are three directions—beyond increased speed—along which Deep Blue could have improved during the year, and its play gave every indication that it improved in all three. Its greater strength over 1996 is much more convincingly explained in these terms than by the modest increase in its look-ahead ability.

Especially interesting is that these improvements correspond to three kinds of chess knowledge possessed by human chess players, enabling them to select good moves after relatively little search (almost certainly never more than 1,000 branches, except when the "book" is followed). Deep Blue's increased speed may be allocated, not to searching deeper but to computing and recognizing more sophisticated patterns, permitting more selective search.

Deep Blue's essential elements can be summarized as:

- Enormous knowledge and databases for the relatively small target domain of chess played on an eight-square × eight-square board. The ratio of knowledge to size of domain is much higher than

Chess aside, what does Deep Blue mean for the future of computers for complex, knowledge-rich tasks and for the role of AI methods in performing these tasks?

in most typical AI applications, such as expert systems.

- A basic strategy not meant to imitate a human but to combine rapid processing with knowledge. The machine is still far behind humans in pattern recognition and its application to intuition and creativity.
- A design that employs its advantage in computational power over human opponents, achieving considerable and growing selectivity in its searches and consequent adaptability in its play.

Chess aside, what does the chess match mean for the future of computers for complex, knowledge-rich tasks and for the role of AI methods in performing these tasks? First, speed alone cannot solve complex problems and must be supplemented with knowledge. Moreover, if there is enough knowledge, the ability to recognize cues and thereby access knowledge associated with particular kinds of situations will gradually replace speed and brute-force search as the main tool for building high-performance systems.

This does not mean computer power is unimportant for solving challenging problems. In the human brain, recognizing patterns in visual and auditory stimuli is perhaps the most expensive computing task performed, sometimes making parallel processing essential. If pattern recognition is central to dealing with complexity, methods are needed for acquiring and evaluating new patterns. Human programming cannot do the job, especially if the program has to be updated continually to incorporate new knowledge. Capabilities must be developed for learning patterns autonomously

Promising Applications

With the prospect of combining the whole range of available AI methods, we can expect many new applications in such areas as molecular dynamics, financial risk assessment, and decision support. Other areas and problems with great promise for the near future and many with ongoing R&D efforts and even successful applications include:

- **Generic categories:**
  - Planning, such as strategic planning and scheduling for domains requiring large knowledge bases
  - Prediction, such as that in natural phenomena (e.g., geophysical, social, and physiological) and technology (e.g., engineering)
  - Pattern recognition, such as that in perceiving and understanding patterns (e.g., image, audio, and general signal processing)
  - Control, such as for machines and parts of the human body (e.g., heartbeat) with precision and sophistication
  - Inference, such as reasoning using massive knowledge and databases (e.g., natural-language processing, intelligent human-machine interfaces, and software engineering)
  - Information retrieval, such as fast and highly filtered extraction of information from large databases
  - Machine learning, such as induction of rules from data, and the scientific discovery of the laws of the natural sciences
- **Application areas:**
  - Engineering, such as designing machinery and developing new materials
  - Geophysics, such as long-range weather forecasting and more accurate prediction of tornadoes and even earthquakes
  - Communication, such as planning networks
  - Transportation, such as creating schedules
  - Robotics, such as autonomous vehicles for hazardous environments
  - Medicine, such as diagnostics and machine-assisted operations
  - Management and finance, such as marketing analysis, production management, scheduling, decision support, economic forecasting, financial analysis, and market prediction and intervention

from information about the task environment. We have considerable experience building programs that learn—reaching back to the late 1950s to Arthur Samuel's pioneering checker-playing program that learned to improve its game by modifying its evaluations with experience. Today, we can learn through "neural nets," adaptive production systems, self-modifying discrimination nets, and genetic algorithms. Experience with Deep Blue suggests a high priority on incorporating such learning capabilities, enabling the system itself to learn from published games and from its own analyses.

If we classify the tasks handled by machines as (I) simple, (II) moderately difficult, and (III) very difficult [1], then at the low end of level I, we have arithmetic computations, simple spelling checks, and straightforward database retrieval. From the upper end of level I through level II, we have most of the practical AI applications today, such as symbolic calculus, expert systems, robotics, and machine vision.

Chess playing against human grandmasters can be regarded as lying toward the upper end of complexity level II. Although the domain is well defined, a high level of play cannot be attained without using sophisticated knowledge and selective search. Although computers are still far behind people in performing in complex everyday environments, they do perform at the level of highly expert humans in domains like chess, using their superior computational power to supplement knowledge and selectivity (see the sidebar "Promising Applications").

What are the prospects for AI solutions to problems at level III complexity? One possibility increasingly examined in recent years is to apply parallel computing methods. Some impressive applications of parallelism today involve speech recognition and vehicle steering.

AI has been used mainly in serial applications for two main reasons: In AI's early days, appropriate parallel machines were not generally available. We still do not understand many things about designing and programming parallel systems to perform tasks involving massive interaction with heavy demand for communication among components and with strong constraints on the order in which tasks may be performed. While most human neural processes appear to be highly parallel, processes that require conscious awareness or going through the narrow bottleneck of short-term memory are largely serial. For example, in symbolic integration, we apply one formula at a time, checking whether it leads toward a solution before taking the next step. AI has focused primarily on tasks with a large serial component. The most notable exceptions are categorical learning tasks involving increasing numbers of applications of statistical and neural net methods.

References

1. Munakata, T. Technical Opinion. Thoughts on Deep Blue vs. Kasparov. *Commun. ACM* *39*, 7 (July 1996), 91–92.

Herbert A. Simon (has@a.gp.cs.cmu.edu) is a professor of computer sciences and psychology at Carnegie-Mellon University.

Toshinori Munakata (munakata@cis.csuohio.edu) is professor of computer science at Cleveland State University.