Intersect, Vol 15, No 1 (2021)

When Moore's Law Killed Chess: How Strategy Games Redefined Intelligence in AI

Cameron Lowry
University of Chicago

The history of AI is a history of games. This piece charts a centennial metamorphosis of machine intelligence in strategy games, from El Ajedrecista in 1912 to Quantum Go in 2020. The purpose of this paper is to highlight modern breakthroughs and disruptions in the way scientists understood machine intelligence, like reinforcement learning and quantum computing. For hundreds of years, chess was associated with intellectual ability, and, in the 20th Century, became the cornerstone of AI research. Now, in the 21st Century, researchers have expanded their work to include the games of Go and shogi, and even revisited chess with new algorithmic approaches previously unattainable. Traditional game-theoretical approaches to computational decision-making have hit a ceiling due to hardware limitations as there are now more permutations of choices and positions than atoms in the universe, way more than a computer could handle. According to the 2019 Stanford AI Index, AI's heavy computational requirement outpaces Moore's Law, doubling every three months rather than two years. This challenged scientists' ability to build intellectual machines. Thus, they grappled with what it meant for a machine to be intelligent and what people could accomplish with intelligent machines.

Introduction

Over one hundred years ago, scientists dreamed of a machine that could play a good game of chess. For centuries, chess was associated with intellectual ability and thus became a criterion for general intelligence in AI as the game "is generally considered to require 'thinking' for skillful play" (Shannon, 1950, p. 2; Esmenger, 2011, p. 9). When algorithms became computationally efficient and mathematically complex, it was able to process more moves than the chess-playing computer's human opponents but at the tradeoff of being less *intellectual*, unable to imitate a human's sub-conscious decision-making processes.

By the 1950s, computers could play a full game of chess without any modifications, such as beginning the round in an end game or excluding the diagonally-moving bishops, for example. Although still amateur level, their programs became more efficient, pursuing promising lines of decisions and rejecting ones with unfavorable outcomes, thus making the computer more productive. With each passing decade, chess-playing computers' skill levels matured as their computer chip speeds increased and evaluation functions improved, allowing them to assess an even greater number of potential moves and decisions per second, even as high as thousands per second. This outpaced human capabilities as computer superintelligence operated leagues beyond what was previously thought possible (Williams, 2017, p. 32).

At the inception of artificial intelligence in the 1950s, Alan Turing, an English computer scientist and mathematician, cautioned that the digital computer "must have an adequate storage capacity as well as working sufficiently fast" (Turing, 1950, p. 441). Hardware limitations incurred by Moore's Law have hindered a theoretically perfect intelligent master for games like chess and Go. Moore's Law stated that computing speed, as measured by the number of transistors in a chip, would double approximately every two years (Mollick, 2006, pg. 1). Tremendous search spaces and exponential combinatorics for potential moves greatly overpowered current capabilities of regular computers to evaluate them. According to the 2019 Stanford AI Index Report, artificial intelligence was now outpacing Moore's Law, doubling every three months rather than every two years (Perrault et al., 2019, p. 65-66). Recent progress in effective algorithms and computer architecture in AI have redefined intelligence for computers, transitioning from brute-force computations to heuristic "thinking."

A Chess-Playing Automaton

The first major development in this area was Leonardo Torres y Quevedo's *El Ajedrecista*. In 1912, Torres built the first autonomous mechanical chess player that, while it could not play a full game, was able to successfully carry out a specific end game strategy involving a white rook and a king (Williams, 2017, p. 30). He presented his contraption at the Paris World Fair in 1914 to the amazement of his audience. Even more impressive, it also recognized when its opponent made illegal moves, flashing a light at them; otherwise, it would continue moving its piece until it checkmated the black king ("Torres and His Remarkable Automatic Devices," 1915, p. 296). One *Scientific American* author, whose name was not documented, smittenly recorded the machine's protests three times in his journalistic piece in the Scientific American ("Torres and His Remarkable Automatic Devices," 1915, p. 296-298). He further wrote that its novelty lay in its ability to select "one possible action in preference to another" (Torres and His Remarkable Automatic Devices," 1915, p.297).

The mechanization of labor and activity characterized the zeitgeist of the Second Industrial Revolution, challenging Torres to automate human intellect. A novel concept, Torres remarked that an even greater feat would constitute a mechanical being that could "imitate, not the simple gestures, but the thoughtful actions of a man, and which can sometimes replace him" (Torres, 1914, p. 89). The automaton could learn from its environment and adapt to varying circumstances around it, responding appropriately to the context it registered. In Torres's mind, self-awareness and discernment, two human-like qualities, distinguished an emerging concept within automata theory from the mundane machines present since the First Industrial Revolution. While his other inventions, like the torpedo, operated self-sufficiently, his theoretical automata eschewed from "meretricious imitation of the human form," an interest that dominated the public's imagination since the *Ajedrecista*'s debut in Paris ("Torres and His Remarkable Automatic Devices," 1915, p. 298).

Later applied to digital computers, this initial conception of machine intelligence synthesized the thoughtfulness of the human brain with the intellectual capability of playing a game of chess. Scientists, philosophers, and mathematicians returned to this definition in the succeeding decades with the advent of programmable computers.

Thinking Machines and Chess

In March 1950, Claude Shannon, an American mathematician and engineer, sparked popular interests in chess-playing artificial intelligence when he proposed a set of algorithms to create a program that played chess. Inspired by Torres, he sought to expand the machine's faculties to play a full game. He hypothesized that if computers could be taught to play chess, they would be capable of activities like translating one language into another and performing symbolic (non-numerical) mathematical operations (Shannon, 1950, p. 1). He presupposed that this game was the key to unlocking numerous opportunities in computer applications within the confines of symbolic AI. If it could play a reasonably skillful game, then one would be forced to confront the possibility that the computer could think like a person (Shannon, 1950, p. 2). The first step, however, was constructing a computer to play a *perfect* game (Shannon, 1950, p. 4).

While such a game was possible in theory, computers were not equipped to handle the sheer complexity and power required to execute this. Mathematically speaking, Shannon noted that there were 10^3 possible outcomes for a white move followed by a black one. Taking a conservative estimate of 40 moves per game, there were $(10^3)^{40}$ or 10^{120} potential variations after the first play. In light of Shannon's estimate, modern researchers raised this approximation to 10^{134} potential variations, calculated from an average of 84 plies per game and 38 moves per plie, bringing the number to 38^{84} or about 10^{134} (Esmenger, 2011, p. 11). The scale of possibilities was overwhelming, even with Shannon's lower estimate. In other words, if one were to enumerate all possible combinations for all possible moves following from a given position at any point in the game through one of three outcomes (win, lose, or draw), they would outnumber atoms in the universe (10^{75}). Moreover, if the computer were to calculate one variation per microsecond, then it would take 10^{90} years to determine the first move of the game (Shannon, 1950, p. 4).

To overcome this impractical calculation, Shannon suggested two evaluation functions to implement in a computer chess player: Type A and Type B. As Nathan Esmenger, a modern historian of artificial intelligence, described this difference:

> The most obvious solution was to reduce the total number of moves that a computer was required to 'look ahead'. This would make the overall decision tree to be evaluated smaller and more manageable, and therefore more amenable to straightforward computational approaches. Shannon called this approach a 'Type-A' solution, and considered it to be a brute-force method that did not accurately reflect the ways in which human beings played chess. He much preferred a 'Type-B' solution that used sophisticated heuristics to trim the decision tree by privileging certain branches over others. Like human grandmasters, Type-B solutions would focus only on the most promising lines of analysis, and would recognize in patterns of positions more general principles of play that would reflect a more truly intelligent approach to the problem of chess (Esmenger, 2011, p. 11).

In short, Type A involved brute force computations and Type B used heuristics to select promising branches in the decision tree and was the most akin to Turing's thinking machine. Even though Shannon preferred Type B due to its parallel to human chess players, computer scientists pursued research into Type A because it was more feasible to implement via the minimax algorithm and alpha-beta searching. Herein lied the difference between playing chess and being intelligent, which sparked decades of debate through the end of the Twentieth Century. Was intelligence the ability to play a good game or was there a *je ne sais quois* to simulating the human mind?

Tackling a similar question, later in 1950, Alan Turing sought to answer his own inquiry "can machines think?" (Turing, 1950, p. 433). As one of the inaugural research questions in this nascent field, Turing and other scientists of the decade associated the ability to think with intelligence; thus, he and his contemporaries pondered the modern equivalent of “can machines be intelligent?” He proposed the Imitation Game, a now-famous thought experiment wherein a human interrogator would ask a series of questions directed at two anonymous beings - one human and one computer - to determine which one was actually the machine. In the instructions, the machine would attempt to deceive the interrogator while the other human would try to help him. To prevent egregious giveaways, answers would either be written or repeated by an intermediary (Turing, 1950, p. 434-435).

In one scenario, Turing imagined a person asking the anonymous computer if it could play chess. A linguistic note, he did not ask if it could play chess well, just that if it could play (i.e., if it knew the rules that governed the game). Following up, he asked about which chess piece it would play in response to a particular chess move. Judging from the context, it appeared to be an endgame analogous to the one *El Ajedrecista* could play, involving a king and a rook. Despite not providing criteria for what was considered a good move, one could assume that only a human could make a logical play. At least, computer theorists believed it to be so. As such, a thinking computer could fool the interrogator if its proposed move seemed human-like. In this hypothetical, the computer moved its rook and checkmated the king.

Although subject to many debates within tech circles, the Imitation Game was the first strategy game that explicitly defined successful machine intelligence. Turing viewed human creativity and strategic analysis as reference points to judge a computer’s performance. This perspective fueled AI fervor for decades to come, pitting people and computers in an intellectual rivalry.

Sunsetting Chess

The Imitation Game’s competitive dynamic had set up the decades-long goal for symbolic chess-playing machines to play against human players. The race to create a digital chess expert drove high hopes for strong artificial intelligence through the 1950s and 1960s. However, Hubert Dreyfus, an American philosopher and notorious critic of artificial intelligence, criticized this seemingly blind optimism. In 1965, Dreyfus published his reflection on the state of AI research regarding games, problem solving, and language translation in his infamous book, *Alchemy and Artificial Intelligence.*

Assessing the stagnation surrounding game research, he recalled H.A. Simon’s bold 1957 prediction at a meeting for Operational Research Society of America that “within ten years a digital computer will be the world’s chess champion” (Dreyfus, 1965, p. 3). However, by the end of the 1960s, computers could not use heuristic decision-making like humans could. Instead, computer scientists programmed them with Type A functions, albeit amateur, due to the feasibility of its implementation.

Dreyfus did not understand how computers could be considered intelligent when progress had been so unfruitful and architecture limitations so obstructing. While the chess-playing algorithms, like minimax and alpha-beta searching, had improved in the 1970s and 80s, scientists and the general public did not care much. It was a scenario of diminishing returns. As efficiency and speed progressed, there were no new major breakthroughs in computational chess (Esmenger, 2011, p. 7). From Dreyfus's perspective, artificial intelligence was nowhere near the intellectual activity as it was to mere monotonous deep searching, a disappointing reflection that once garnered ridicule but soon became accepted by the wider community.

Regarding the limitations of digital computers playing chess, he said one of the benchmarks for a system to equal human performance would lie in its ability to recall information from the fringes of consciousness, where it is neither disregarded nor at the forefront of thought (Dreyfus, 1965, p. 45-46). Dreyfus discussed this to address supposed shortcuts in chess heuristics, something he considered a fallacious idea. Some of his other criteria involved taking "into account the context" and "distinguishing the essential from the inessential features of a particular instance of a pattern" (Dreyfus, 1965, p. 45-46). Likewise, people were capable of recognizing patterns in ambiguous and difficult conditions, but "work in pattern recognition has not progressed beyond the laborious recognition of a few simple patterns in situations which severely limit variation" (Dreyfus, 1965, p. 46).

After years of disappointment and disillusionment, it was believed that there could be no new progress in artificial intelligence, bringing about the first AI Winter in the 1970s, a period when funding and interest in AI research declined (Paine, 2005). Marginal achievements could not excite the broader public anymore.

DeepBlue, the Digital Grandmaster

By the end of the 1970s, computer engineers found that a major way to advance brute-force algorithms was through processing speed, a feature that they could easily improve (Hsu et al., 1995, p. 240). Hardware performance being a limitation in previous decades, it soon lended itself to benefit research in this topic area. Starting in the late 1980s, IBM challenged itself to construct a successful chess algorithm in conjunction with its prowess in microchip development.

Once considered a Sisyphean task, IBM constructed several chess-playing programs, beginning with Deep Thought in 1988. That year, it differentiated itself as the first computer to achieve grandmaster status (Hsu et al., 1995, p. 240). A year later, IBM set up a tournament with Deep Thought against Gary Kasparov, the former world chess champion. The machine lost, but it inspired development for a better computer. In 1991, IBM researchers created Deep Thought 2.

Aware that hardware was the key, they enhanced the single chip chess processor to fill in “long range chess knowledge gaps” (Hsu et al., 1995, p. 241). They also gave it a larger RAM and a newly written search software (Hsu et al., 2002, p. 58). An immediate precursor to the DeepBlue computers, Deep Thought 2 competed publicly until it retired in 1995.

IBM’s chess computer iterations led to the creation of DeepBlue I, which also lost to Kasparov in 1996. Throughout the remainder of that year, researchers rectified deficiencies in this computer in order to build a stronger computer, DeepBlue II. Some major changes included enhanced chess chip designs, repetition detection, doubling the number of chips, and remaking its debugging software (Hsu et al., 2002, p. 59). Trained on the 1996 Kasparov match, DeepBlue II’s evaluation function was built into its hardware, simplifying the task of programming it and allowed for greater flexibility in improving it (Hsu et al., 2002, p. 61).

In 1997, DeepBlue II challenged Kasparov at the Equitable Center in New York. Decades of little progress were soon overlooked by this globally televised match. A final showdown between man and machine, the grandmaster won the first game, the computer won the second, and the last three rounds ended in a draw. Over a course of 80 years, the world was finally presented with the first computer to defeat a human world chess champion in the game, a more significant accomplishment than just obtaining grandmaster status.

While a major feat, the computer was not capable of comprehending the moves it played; it was only able to compute them faster than its opponent could. Advances in computer chess stagnated when the only progress was faster brute-force algorithms, straying from the field’s original thesis to make the computer smarter. Kasparov further vented that he did not “know what the computer did wrong or right” nor could he understand its “ability to evaluate those positions” (Weber, 1997). IBM’s DeepBlue II was not thinking, which frustrated Kasparov. Human grandmasters did not think of these strategy games in terms of data but rather in heuristics. If implemented computationally, it would be akin to Shannon’s Type B approach or Turing’s thinking machine.

Like its predecessors, DeepBlue II was an algorithmic black box but one that its team sought to enhance. They knew that they “could make the computer faster, so they concentrated on making it smarter, [but] it was not entirely clear how to do that” (Weber, 1997). This begged the question: what *does it mean to make the computer smarter*? If that was the agenda throughout AI development, then two schools of thought had unintentionally come forth. One focused on imitating intellect and another focused on winning chess, even in a roboting, mundane way. To re-examine the Turing question, it was clear that it was not enough for a thinking computer to play chess moves. An intelligent computer would understand them.

And Then There Was Go

In a post-chess world, attention turned towards Go. Like chess, Go was once perceived as a game that computers could not beat since it was much more complex (Silver et al., 2016, p. 484). Traditional brute-force AI algorithms that had dominated the research space would be impossible for this game, unlike in chess (Silver & Hassabis, 2016). Facing the same obstacles from Moore's Law as DeepBlue II and other chess computers, researchers in the early 2000s turned to more creative methods to revisit Turing's question.

From the start of the new millenium, Go became the new cornerstone of AI research. More sophisticated than chess, Go was the ideal game for intuition, not forward-looking predictions:

> The upshot is that, unlike in chess, players—whether human or machine—can't look ahead to the ultimate outcome of each potential move. The top players play by intuition, not raw calculation. "Good positions look good," Hassabis says. "It seems to follow some kind of aesthetic. That's why it has been such a fascinating game for thousands of years" (Metz, 2016).

With its even larger search space, brute force tree searching was unable to evaluate all possible moves (Qiao et al., 2020, p. 1). If chess had an average number of moves of 40 per game, then Go had 200 (Metz, 2016). Unlike chess, Go's complexity would allow scientists to be extra certain that the decisions made by the computer would better simulate actual intelligence.

When DeepMind built AlphaGo in 2015, it revolutionized the way the public viewed AI as it was the first time that a computer won the game against a professional player, taking the lead in all five rounds. When it went on to compete against Lee Sedol in March 2016, the team boasted that its "search algorithm is much more human-like than previous approaches," opting to go for a heuristic Monte Carlo search tree than a brute-force method (Metz, 2016). Considered a superintelligent move and a pinnacle moment in the tournament, AlphaGo's infamous *Move 37* confused Sedol as it was an irrational choice that no human would have made.

What's more telling was that, like the scientists before them, they not only wanted to teach computers to play a strategy game well, but to also beat human experts at them. In this way, the computer would become a superintelligent machine, drawing criticism as to how intelligent it could be if it did not understand what it was doing. This question had been masked by decades of competitiveness between people and computers. Go-playing computers once again faced the same shortcomings as chess: being an expert at a game did not give it the human-like intelligence that computer scientists had hoped for.

In October 2017, DeepMind rolled out with AlphaGo Zero, utilizing reinforcement learning to play Go without previous knowledge or data about it, being trained on only its own plays. This was a radical shift from the symbolic logic approach that governed chess research during the latter half of the Twentieth Century. Whereas AlphaGo trained for several months to learn Go, this version of the program outperformed it after 36 hours (Silver et al., 2017, p. 8). Moreover, in just three days, "starting tabula rasa, AlphaGo Zero was able to rediscover much of this Go knowledge" without being "constrained by the limits of human knowledge" (Silver et al., 2017, p. 14; Silver & Hassabis, 2017).

Just a couple months later, in December 2017, DeepMind debuted AlphaZero, a more generalized version of AlphaGo Zero, wherein it could play other strategy games like shogi and chess in addition to Go (Silver et al., 2017, p. 2). Different from its predecessors, it was not designed to play any particular game but was taught the basic rules of each game "with no other strategies or tactics" (Vincent, 2017). AlphaZero involved less computations and evaluation functions than similar programs. As a point of comparison, it "searches just 80 thousand positions per second in chess and 40 thousand in shogi, compared to 70 million for Stockfish and 35 million for Elmo." Instead, it took a more "human-like approach to search, as originally proposed by Shannon" and utilized a deep neural network to hone in on promising sequences of moves (Silver et al., 2017, p. 5). As it turned out, in addition to the Monte Carlo tree search, the key to achieving Shannon's Type B approach was neural networks and reinforcement learning. By filtering for the most optimal patterns for their play, these algorithms utilize computational heuristics to play chess and other similar games as Shannon envisioned almost 50 years prior.

A Cooperative Imitation Game

When strategy games, like chess and Go, returned to the limelight of AI research, the next focus shifted to the AI algorithms themselves. In 2020, researchers at Microsoft, Cornell University, and University of Toronto developed the Maia chess engine, which could understand the decisions it made in a given game and emulate its decision-making like a human's. They claimed that a "crucial step in bridging this gap between human and artificial intelligence is modeling the granular actions that constitute human behavior rather than simply matching aggregate human performance" (McIlroy-Young et al., 2020, p. 1677).

Herein one is presented with yet another definition of machine intelligence. It was not the outcome that determined it but rather the play-by-play decisions it presented in a chess match. Moreover, to align artificial intelligence with human behavior, they personalized it to play at different, specific skill levels (McIlroy-Young et al., 2020). There were already a plethora of chess engines, like Stockfish and Leela (the open-source version of AlphaZero), that can beat world grandmasters, but it was no longer fun to play a game in which one consistently lost. For the research team, a bigger challenge was for the computer to cooperate with a person and essentially teach them how to play well and improve (McIlroy-Young et al., 2020). It could even predict a person's decisions at each individual skill level, paving the way for collaboration between people and artificial intelligence (McIlroy-Young et al., 2020). Maia was not only more accurate than Leela, but it was also explainable, so researchers could understand how and why it chose to play particular moves in chess. This challenged the fallacious dichotomy between accuracy and explainability in artificial intelligence. Instead of viewing chess-playing computers as a superintelligent opponent, the team shifted its perception to be seen as a teacher.

While the words *superintelligent* and *intelligent* were contested when it came to describing artificial intelligence, this team ultimately shaped its potential to become human-like. Maia was a revisitation of the Torres-ian view of machine intelligence, in which the automaton - or, rather, the computer - imitated and simulated human thought, decisions, and intelligence. It even returned to Turing's Imitation Game, where a computer was thought to be intelligent if it could trick a person into thinking it was a human, the difference being that Maia would not deceive them but instead help them. Imagine if in the Imitation Game, the computer told the interrogator how to save its king from a doomed endgame. That was the new philosophy of cooperative computational chess.

Quantum Computing and Games

While the door seemed to close on strategic games in classical computing, the rise in popularity of quantum computers throughout the 2010s had opened up research areas into quantum chess and quantum Go. One major difference between quantum computers and traditional computers was that the former did not follow Moore's Law, so it could not be restrained by it. Rather, they followed Rose's Law, which stated that the number of qubits (quantum bits) doubled every year. This is a much steeper rate of growth, compounding faster than Moore's Law. It is "more than 10^8 times faster" than a classical computer, allowing it to perform optimization problems more efficiently (Neven, 2015). As such, quantum computers would be suitable for optimizations in strategy games.

In 2020, around the same time that American researchers investigated a human-like chess program, Chinese researchers looked into Go, or rather Quantum Go. While the game had been used as a testbed for artificial intelligence, it was not the most difficult game to teach a machine learning algorithm since it was deterministic and had perfect information; as such, it could *easily* search possible moves (Qiao et al., 2020, p. 1). Hitting a ceiling with progress and public attention, the "community moved interest to nondeterministic and imperfect information games" (Qiao et al., 2020, p. 1). In an environment of asymmetric information, players had to guess the other player's knowledge and deal with uncontrolled randomness, as in games like Poker or Mahjong, which made them the ideal experiments for advanced machine learning algorithms (Qiao et al., 2020, p. 1).

With its quantum computer architecture, Quantum Go incorporated randomness into the game and could "cover a wide range of game difficulties," just as Maia could with different skill levels in chess (Qiao et al., 2020, 7). This distinguished the quantum game-player from other imperfect information and nondeterministic games. It functioned outside the assumptions of classical game theory that guided games like chess and Go. As such, it became an optimal benchmark for artificial intelligence. Since humans could handle unpredictable circumstances, quantum computing could possibly be able to answer the quintessential question *can machines think*. While resolving Dreyfus's criticisms, quantum artificial intelligence could handle information on the fringes of consciousness and take context into account, such as skill level and uncontrolled randomness.

Conclusion

After years of stagnation during the AI winter, computer scientists presented the world with thinking machines in the realms of chess and Go. AI capabilities overcame the restrictions outlined by Moore's Law, in which computing speed doubles every two years, by implementing heuristic-based approaches to computational decision-making. Even though there were major advancements in reinforcement learning, heuristic tree searching, and quantum computing being applied to strategy games, there is a long way to go to define machine intelligence, let alone have games be a proxy for it. With new algorithms to model artificial intelligence, scientists still need to grapple with what it means for a computer to be intelligent. Correlation between strategy games and intelligence does not always equal causation. Nevertheless, with games at the forefront of research, scientists were able to make great breakthroughs in artificial intelligence, shifting the paradigms from symbolic logic to deep learning and thus bringing the world one step closer to simulating human intellect by one dimension.

References

Dreyfus, H. (1965). *Alchemy and Artificial Intelligence*. RAND Corporation. Retrieved from https://www.rand.org/pubs/papers/P3244.html

Esmenger, N. (2011). Is Chess the Drosophila of Artificial Intelligence? A Social History of An algorithm. *Social Studies of Science* 42(1), 5 - 30. Retrieved from https://journals.sagepub.com/doi/pdf/10.1177/0306312711424596

Hsu, F., Campbell, M. S., & Joseph Hoane Jr., A. (1995). Deep Blue System Overview. *Association for Computing Machinery: ICS '95: Proceedings of the 9th International Conference on Supercomputing.* 240 - 244. Retrieved from https://web.archive.org/web/20181017043132/http://www.top-5000.nl/ps/Deep blue system overview.pdf

Hsu, F., Campbell, M. S., & Joseph Hoane Jr., A. (2002). Deep Blue. *Artificial Intelligence 134*(1-2) (2002): 57 - 83. Retrieved from https://www.sciencedirect.com/science/article/pii/S0004370201001291

McIlroy-Young, R., Sen, S., Kleinberg, J., & Anderson, A. (2020). Aligning Superhuman AI with Human Behavior: Chess as a Model System. *Association for Computing Machinery: KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1677 - 1687. Retrieved from https://dl.acm.org/doi/10.1145/3394486.3403219

McIlroy-Young, R., Sen, S., Kleinberg, J., & Anderson, A. (2020). The Human Side of Chess for AI. *Microsoft Research Blog*. Retrieved from https://www.microsoft.com/en-us/research/blog/the-human-side-of-ai-for-chess/.

Metz, C. The Rise of Artificial Intelligence and the End of Code. (2016). *Wired*. Retrieved from https://www.wired.com/2016/05/google-alpha-go-ai/.

Mollick, E. (2006). Establishing Moore's Law. *IEE Annals of the History of Computing 28*(3). 62-75. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1677462

Neven, H. (2015). When can Quantum Annealing Win? *Google AI Blog*. Retrieved from https://ai.googleblog.com/2015/12/when-can-quantum-annealing-win.html

Paine, J. (2005). An AI Alphabet. *AI Expert Newsletter*. Retrieved from https://web.archive.org/web/20131109201636/http://www.ainewsletter.com/newsletters/aix_0501.htm#w

Perrault, R., Shoham, Y., Brynjolfsson, E., Clark, J., Etchemendy, J., Grosz, B., Lyons, T., Manyika, J., Mishra, S., & Niebles, J.C. (2019). The AI Index 2019 Annual Report. *AI Index Steering Committee, Human-Centered AI Institute, Stanford University.* 65-66. Retrieved from https://hai.stanford.edu/sites/default/files/ai_index_2019_report.pdf