Where Does AlphaGo Go: From Church-Turing Thesis to AlphaGo Thesis and Beyond

Fei-Yue Wang, *Fellow, IEEE,* Jun Jason Zhang, *Senior Member, IEEE,* Xinhu Zheng, *Student Member, IEEE,* Xiao Wang, Yong Yuan, Xiaoxiao Dai, *Student Member, IEEE,* Jie Zhang, Liuqing Yang, *Fellow, IEEE*

Abstract—An investigation on the impact and significance of the AlphaGo vs. Lee Sedol Go match is conducted, and concludes with a conjecture of the AlphaGo Thesis and its extension in accordance with the Church-Turing Thesis in the history of computing. It is postulated that the architecture and method utilized by the AlphaGo program provide an engineering solution for tackling issues in complexity and intelligence. Specifically, the AlphaGo Thesis implies that any effective procedure for hard decision problems such as NP-hard can be implemented with AlphaGo-like approach. Deep rule-based networks are proposed in attempt to establish an understandable structure for deep neural networks in deep learning. The success of AlphaGo and corresponding thesis ensure the technical soundness of the parallel intelligence approach for intelligent control and management of complex systems and knowledge automation.

Index Terms—ACP, AlphaGo, AlphaGo Thesis, Church-Turing Thesis, deep learning, deep neural networks, deep rule-based networks, knowledge automation, parallel intelligence, parallel control, parallel management.

I. INTRODUCTION

THE match of AlphaGo vs. Lee Sedol is a history making event and a milestone in the quest of artificial intelligence (AI). The computer Go program AlphaGo by DeepMind has won 4:1 in a five game match against one of the world's best

Fei-Yue Wang is with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences (SKL-MCCS, CASIA), Beijing 100190, China, and also with Research Center of Computational Experiments and Parallel Systems, The National University of Defense Technology, Changsha 410073, China (e-mail: feiyue.wang@ia.ac.cn).

Jun Jason Zhang and Xiaoxiao Dai are with the Department of Electrical and Computer Engineering, Ritchie School of Engineering and Computer Science, University of Denver, Denver, CO 80210, USA (e-mail: jun.zhang@du.edu, xiaoxiao.dai@du.edu).

Xinhu Zheng is with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55414, USA (e-mail: zheng473@umn.edu).

Xiao Wang, Yong Yuan and Jie Zhang are with the Qingdao Academy of Intelligent Industries (QAII), Qingdao, Shandong, China, and SKL-MCCS, CASIA, Beijing 10019, China (e-mail: x.wang@ia.ac.cn, yong.yuan@ia.ac.cn, jie.zhang@ia.ac.cn).

Liuqing Yang is with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA (e-mail: lqyang@engr.colostate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

players, Lee Sedol, from Korea. The victory has come considerably sooner than anyone has expected and has astonished many in the AI field. Nolan Bushnell, the founder of Atari and a Go Guru himself, was so impressed by AlphaGo's feat: "Go is the most important game in my life", he said, "It is the only game that truly balances the left and right sides of the brain. The fact that it has now yielded to computer technology is massively important"[1]. The defeat over humanity by a machine has also generated huge public interests in AI technology around the world, especially in China, Korea, U.S., and U.K. To many people, IT has a new meaning from this moment: it stands not just for Information Technology or Industrial Technology, it is Intelligent Technology now, and the age of new IT is coming[2].
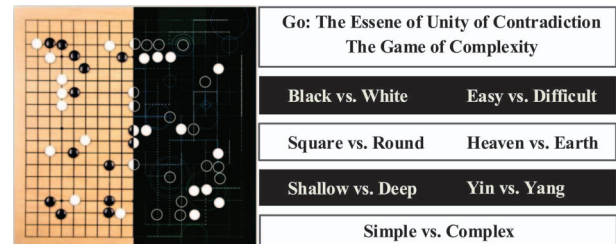

Fig. 1. Go: A game of complexity and a symbol for unity of contradiction.

The game of Go is originated from ancient China and its earliest documentation dated back to the Western Zhou Dynasty (1046 BC to 771 BC). It was introduced to the Korean Peninsula and Japan during 420 AD to 589 AD and 618 AD to 907 AD, respectively. The rule of Go is extremely simple, however, its complexity is prohibitively high. The number of theoretically possible games is in the order of 10^{700}[3], which defeats any attempt of applying only brute-force computing, and demands real intelligence in Go programs against human players. Furthermore, the victory of Go relies heavily on fuzzy and objective interpretation of the overall game situations, such as *thickness*, *liberty* and *potential*, which creates tremendous obstacles for computers to succeed. This is the reason why Go is so challenging and attractive to AI researchers.

However, very few studies and little progress in AI for Go have been made before the 1990 s[4−7]. Prior to 2015, the best achievement Go programs could obtain is merely an *amateur dan* level[8−12]. A breakthrough was made by AlphaGo in October 2015 after it defeated 5:0 the European Go champion, Fan Hui, of *professional two dan*[8]. Today many top players believe that AlphaGo is above *nine dan*, the highest profession-

al level in Go.

Here we would like to consider the impact and significance of AlphaGo's success from a different perspective, i.e., in terms of the fundamental issue of complexity and intelligence. Similar to the Church-Turing Thesis in early history of computing machinery, we believe that an AlphaGo Thesis can be postulated for intelligent technology, and it will play an important role for intelligent industries and smart societies.

II. Impact of AlphaGo: The Coming Era of New IT

In terms of technical innovation, AlphaGo did not create new AI concepts or methods. However, its greatest contribution lies in the massive integration and implementation of recent data-driven AI approaches, especially deep learning (DL) and advanced search techniques, and demonstrates to the world the power of these technological advancements. For this reason, the AlphaGo's victory and achievement is beyond the technology and more of psychological nature.

Deep neural networks (DNN) based DL has played a vital role in AlphaGo's success, but, the features learned automatically by DNN can not be understood intuitively by human. How to interpret the feature spaces generated by the DL process? What are their characteristics in terms of reliability, stability and convergence?

To a large extend, the incomprehensibility of the features and their spaces might be precisely the reason why the AlphaGo possesses the intelligence beating human players in Go games. This also verifies the belief of many AI experts that the intelligence must emerge from the process of computing and interacting. As stated by one of AI's Founding Fathers, the late MIT professor Marvin Minsky, "What magical trick makes us intelligent? The trick is that there is no trick. The power of intelligence stems from our vast diversity, not from any single perfect principle"[13].

Compared with other Go programs, the AlphaGo is unique in its ability of relating the immediate action with the final outcome of the game through its value network, providing a mechanism for closed-loop feedback in decision-making. This is very similar to the adaptive dynamic programming (ADP) developed in control[14−16], except that normal neural networks instead of DNN are used in ADP.

In addition, the data generated and improvements obtained by reinforcement learning through a huge amount of self-play games are extremely impressive. Actually, it is possible that the number of self-play games by AlphaGo is even larger than the number of games played by the entire human race since the creation of Go, which is unimaginable to any single human being in the past as well as in the future[17].

Obviously, the AlphaGo approach can be applied in many other areas other than the game of Go. Clearly, after the "old" IT (Industrial Technology) and the "past" IT (Information Technology), the human society will enter the era of the "new" IT (Intelligent Technology) represented mainly by AI and robotics. Therefore, AlphaGo goes to an updated state of IT, i.e.,

$$IT = Old\ IT + Past\ IT + New\ IT. \qquad (1)$$

Correspondingly, we have to be moved from Big Materials, Big Manufacturing and Big Markets with Cyber-Physical Systems (CPS), to Big Data, Big Computing and Big Decision with Cyber-Physical-Social Systems (CPSS).

However, to complete this transition, we need a paradigm shift in our way of thinking. In the old IT time, we need the confirmation by casuality from science for our action. In the era of new IT, the confirmation by association from data and computing is sufficient before making decisions. Such computational thinking supported by New IT will lead to a computational culture and eventually pave the road to a computational and smart society.
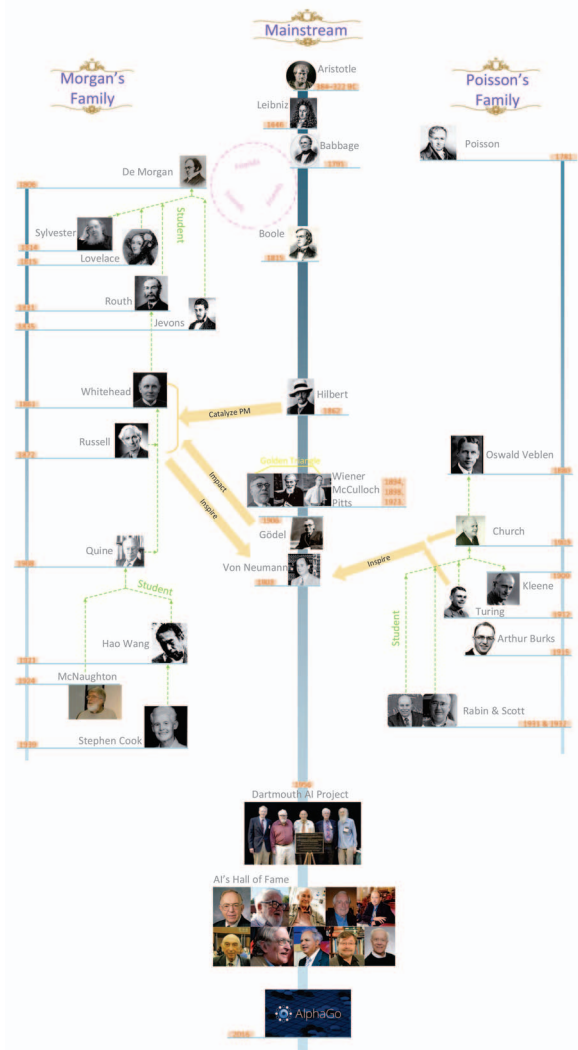

Fig. 2. Historical milestones in AI: A retrospective from two academic families.

III. Go vs. Complexity and Intelligence: Historical Retrospective

Starting from Aristotle's formal logic, Leibniz's binary number and symbolic thought, Boole's algebra and laws of thought, as well as Babbage's difference and analytic engines, many outstanding scholars and scientists before the 20th century had been striving for machines with human intelligence. Over 100 years ago, new efforts have been inspired and intensi-

fied by Hilbert's problems and program for axiomatization and mechanization of mathematics. Among them, Russell and Whitehead's *Principia Mathematica* (*PM*) was the most successful one. Under direct and heavy influence of the *PM*, Wiener, McCulloch and Pitts worked individually as well as jointly for development of cybernetics, brain modeling and artificial neurons, which laid the foundation of cognitive science, artificial intelligence and computational intelligence, especially initiated the road to deep learning that has been employed by AlphaGo[18].

However, in 1930, Gödel's incompleteness theorems smashed Hilbert's grand plan as well as Russell and Whitehead's vision in *PM* for mathematics. In 1936, Turing reformulated Gödel's results on the limits of proof and computation using his simple device (Turing machine), and proved that Turing machine is able to perform any conceivable computation for all representable programs. In this process, Turing also solved Hilbert's Entscheidungs problem, the decision problem, by showing that the halting problem for Turing machine is *undecidable*. Turing's work was published shortly after Church's equivalent work using his more elaborated λ-*calculus*. The concepts of Turing's computability and Church's effective calculability led to the Church-Turing Thesis, postulating that all representable functions can be calculated by Turing machines. Inspired by the Church-Turing Thesis, and assisted by some of Wiener's ideas[18], von Neumann designed Electronic Discrete Variable Automatic Computer (EDVAC), one of the earliest electronic computers, and established the von Neumann architecture which is still used by modern computers today.

In 1956, Wiener's colleagues at MIT, especially, Oliver Selfridge, one of Wiener's graduate students, helped with organizing the first workshop on cognitive science in MIT, and later in the summer, he participated in the Dartmouth conference that officially launched the AI field. Although Wiener did not personally take part in those activities, he still exerted a great influence[18]. Unfortunately, the idea of feedback, central to his cybernetics, does not take a significant role in AI.

Very soon, the quest of intelligence met the complexity of decision-making. And people found that Turing machine is too unrealistic to be useful for modeling any practical decision-making process[19−22]. In the late 1950s and entire 1960s, research efforts were shifted from Turing machines to finite automata and net[21−23], and significant progress had been made on complexity of decision-making problems with automata. In 1959, Rabin and Scott pioneered the field of computational complexity for decision problems with non-deterministic finite state automata[23−24]. For this work, they received A.M. Turing award in 1976. In 1971, Cook advanced their work on computational complexity and laid the foundation for the theory of NP-Completeness and received the A.M. Turing Award in 1982[25]. Although these studies had advanced our understanding of complexity significantly, not much progress has been made in actual computation in the quest for optimization and intelligence. In the attempt to deal with the complexity of constructing intelligent machines, the concept of ϵ-computational complexity based on entropy and

Kolmogorov's measure was introduced by considering specific computational resource available[26], however, no further investigation had been made since.

In the 1990s, significant efforts had been launched in dealing with NP-hard or NP-complete problems using neural networks, evolutionary computing, and other methods in computational intelligence[27−31]. However, these investigations revealed that even finding approximate solutions to NP-hard problems by neural networks is still hard.

From the perspective of the historical development of AI, many successes of AlphaGo is necessity of fortuity, and it is the natural result combining the efforts and wisdom from talents from various fields. AlphaGo utilizes two types of deep learning networks, the policy-networks and value-networks, for creating a closed-loop evaluation mechanism for the overall situation as well as the optimization for the current action, which is the evidence of Wiener's vision on feedback to his cybernetics. AlphaGo utilizes deep neural networks which can be dated back to the era of Wiener, McCulloch and Pitts. The deep learning is a great advancement on top of the achievements on neural networks from 1950s to 1990s, and also indicates that neural networks need to be "deep" in order to be effective for fighting decision-making complexity problems. AlphaGo uses heuristic search algorithms, the Monte Carlo tree search, for searching its strategic space, which corresponds to the complexity theory and more recent computational research efforts in tackling complex problems. One more essential element in AlphaGo that should not be overlooked is the CPU and GPU based large-scale and parallel computing. It is the latest development reflecting the nature of Church-Turing thesis, and provides the foundation for bringing the deep learning, search and feedback mechanism to the real world. With the massive integration of all the elements from the AI history, AlphaGo succeeded in possessing intelligence in Go, however, one question remains, i.e., whether AlphaGo's success can be replicated in other AI domains, and even eventually leads to the coming of the true era of AI?

IV. FROM CHURCH-TURING THESIS TO ALPHAGO THESIS AND BEYOND

Gödel's incompleteness theorems completely smashed the hope and denied the mechanism that a system can derive generic mathematical truth, which was postulated by Hilbert, Russell and Whitehead. Although Turing provides a "machine" that can perform any conceivable computation for all representable programs, the computational procedure's "halting problem", or the decision problem, is still undecidable. The subsequent complexity research provided us a deep understanding on the structures of decision complexity problems, such as NP, NP-Complete, co-NP, etc., however, in a general sense, enumeration, or "brute force" computation, which is powerless in solving complex problems as in the case of "combinatorial explosion" and "curse of dimensionality", seems to be the only measure for search and decision-making. Holland's genetic algorithms (GA) and the later computational methods, such as the evolutionary computation (EC), received considerable achievements in decision optimization and complexity

problems, but, until now, they have not exhibited the ability in solving problems with the complexity that reaches the level of intelligence required by AI.

Complexity is the source of intelligence. Pierre Teilhard de Chardin views evolution as a process resulted from increasing complexity[32], and with such argument, intelligence is computation with increasing complexity! The nature of complexity is the unity of contradiction, and Go is the symbol for unity of contradiction, making it a perfect game of complexity (Fig. 1). Perhaps, we can find out a general solution to all practical computational complexity problems through analyzing the structure and algorithms employed by AlphaGo. And this is the objective for establishing the AlphaGo Thesis[17].

Specifically, the conjecture of the AlphaGo Thesis states:

The AlphaGo Thesis. A decision problem for intelligence is tractable by a human being with limited resources if and only if it is tractable by an AlphaGo-like program.

Here, AlphaGo-like programs are computer programs that use similar system architectures, methods and resources as in AlphaGo. Deep learning (DL) and Monte Carlo tree search (MCTS) are key to AlphaGo, where deep neural networks provide sufficient numbers of parameters for AlphaGo to conduct complex computation for human-level intelligence on specific tasks. This is achieved by its reduction and learning capacities in decision space and searching processes.

In fact, if the number of adjustable parameters is large enough, and the methods and procedures for learning and searching are effective, there will be no restriction on using only DNN, DL or MCTS. One could certainly work out other equivalent models and methods to duplicate the success of AlphaGo. Therefore, we would be able to establish the Extended AlphaGo Thesis.

The Extended AlphaGo Thesis. A decision problem is tractable by a human being with limited resources if and only if it is tractable by a network-like system with sufficient numbers of adjustable parameters.

More discussions on this can be found in [17]. In the next section, we will propose a deep network that can be conveniently understood by a human being, but might be equally useful for decision-making as DNN for DL.

V. DEEP RULE-BASED NETWORKS FOR DECISION AND EVALUATION

AlphaGo utilizes DNN to construct its policy-networks and value-networks, however, the meaning of the feature extraction procedure by DNN is difficult to understand. Earlier works in [33-35] provided a method which embeds knowledge architecture into neural networks. It utilized fuzzy logic and genetic evolutionary computing to derive corresponding learning algorithms and methods for eliminating inapplicable rules or generating new rules[11−12, 36]. Based on this, the deep rule-based network is provided in Fig. 3, where Classification Network classifies input vector x, and generates M patterns; the Association Network associates the patterns with K decision rules; through the Deduction Network, N patterns are designated as the input to the next stage; finally, the Generation Network generates output vector y. In this way, we do not have to specify how to implement the Classification, Association, Deduction and Generation Networks, and they might be convolutional neural network (CNN), deep belief network (DBN), stack autoencoder (SAE), recursive neural network (RNN), or other forms of networks, however, the nodes S_m, R_k and U_n can be interpreted as the following set of Fuzzy Logic Rules:

$$\text{Rule-}\mathbf{r} : \text{If } x \text{ is } S_r, \text{ then } y \text{ is } U_r, 1 \leq r \leq M \times N \qquad (2)$$

where $S_r \subset \{S_1, \ldots, S_M\}$ and $U_r \subset \{U_1, \ldots, U_N\}$ denotes the input pattern and output pattern in rule $\mathbf{r}$, respectively.

In practical applications, the DRN shown in Fig. 3 can be implemented by complex hardware or cloud computing, while the rule set in (2), the fuzzy logic decision rules, can be implemented by low-cost and simple hardware, and the combination of these two forms represents the design philosophy of "Simple Locally, Complex Remotely" or "Simple Online, Complex Offline". The detailed description and corresponding learning algorithm of DRN are to be discussed in our other articles.
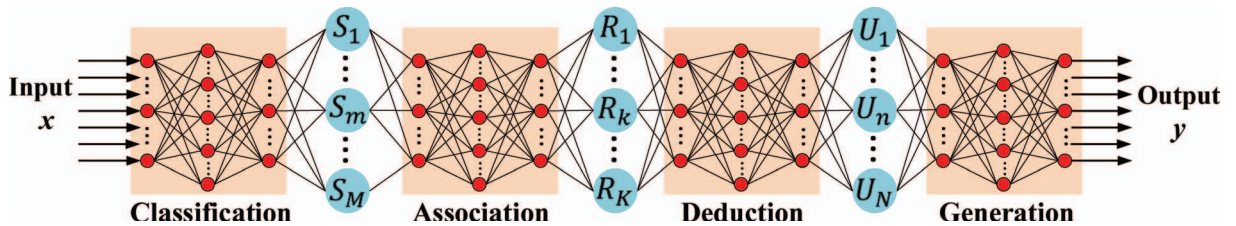

Fig. 3.   Deep rule-based networks: Building a understandable decision language for deep networks.

VI. PARALLEL INTELLIGENCE: A DATA-DRIVEN APPROACH FOR INTELLIGENT DECISION-MAKING

Fig. 4 demonstrates the architecture of parallel control and management for complex systems developed over a decade ago by Wang[34, 37]. AlphaGo provides new realistic and effective support to the architecture, and based on this, a new technical path is paved for parallel intelligence and data-driven intelligent decision. In parallel intelligent systems, every physical process corresponds to one or more software-defined systems (SDS). A SDS can be a DNN, DRN or other artificial systems. The construction process of SDS is the process of learning, mainly based on existing knowledge, data, experience or even intuition. With SDS, computational experiments can be conducted, i.e., self-play, self-running or self-operations, various self-X, and from the SDS computational experiments, huge amounts of data are generated. The data are then used for reinforcement learning to enhance intelligence and decision

capabilities, meanwhile, the decisions are evaluated against various conditions. In the end, the physical processes and systems interact with the software-defined processes and systems, forming a closed-loop feedback decision-making process to control and manage the complex situations. This is the core concept of the ACP-based parallel intelligent systems, which is a more generalized form of the AlphaGo program. In ACP, "A" stands for "artificial systems", which works for building complex models; "C" denotes "computational experiments",

which aims at accurate analysis and reliable evaluation; and "P" represents "parallel execution", which targets at innovative decision-making. And they correspond to descriptive, predictive and prescriptive learning, analytics and decision-making. As indicated in Fig. 5, such parallel intelligence can be used in three modes of operations: 1) Learning and training, 2) Experiment and evaluation, 3) Control and management. The details of the ACP approach and parallel intelligence are depicted as following.
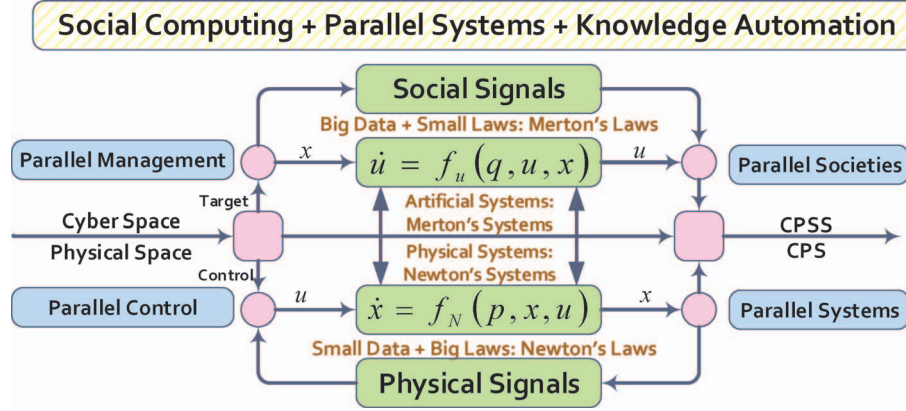

Fig. 4. Parallel intelligence for decision-making: An architecture of parallel control and management for complex systems.

A. ACP Approach and AlphaGo

The core philosophy of ACP is that, through combining "artificial systems", "computational experiments" and "parallel execution", one can turn the virtual artificial space, i.e., the "cyberspace", into the another space for solving complexity problems. The virtual artificial space and the real physical space construct the "complex space" for solving complex system problems. The most recent development of new IT technology, such as high-performance and cloud computing, deep learning, and Internet of Things, provides a technical foundation for the ACP approach. In essence, the ACP approach aims to establish both the "virtual" and "real" parallel systems of a complex system, and both of them are utilized to solve complex system problems through quantifiable and implementable real-time computing and interacting.

In summary, the ACP approach is consist of three major steps. 1) Using artificial systems to model complex systems; 2) Using computational experiments to evaluate complex system; and 3) Interacting the real physical system with the virtual artificial system, and through the virtual-real system interaction, realizing effective control and management over the complex system.

In a sense, the ACP approach solves the "paradox of scientific methods" in complex system's "scientific solutions". In most complex systems, due to high complexity, conducting experiments is infeasible. In most cases, one has to rely on the results, i.e., the output from the complex system, for evaluating the solution. However, "scientific solutions" need to satisfy two conditions: triable and repeatable. In a complex system that involves human and societies, not being triable is a major issue. This is due to multi-faceted of reasons of prohibitive costs, legal restriction, ethics, and most importantly, impos-

sible to have unchanged experiment conditions. The above inevitably results in the "paradox of scientific methods" in complex system's "scientific solutions".

As a result, the ACP pursues the sub-optimal approach of "computational experiments". The computational experiments substitute the simulations for physical systems when simulations are not feasible. In this way, the process of solving to complex system issues becomes controllable, observable and repeatable, and thus the solution also becomes triable and repeatable, which meets the basic requirements of scientific methods.

Using the ACP approach to interpret AlphaGo, we provide the following remarks. AlphaGo uses deep learning from Big Data of existing Go games to construct its software-defined Go games. It uses computational experiments, i.e., self-play games, for further exercising on possible situations of the games. Finally, in the match vs. Lee, AlphaGo searches its strategy space in the virtual games. After that AlphaGo applies the search result in the real game. After one iteration with the real physical world (one step), evaluation of the strategy is assessed in real-time for determining its next action.

B. AlphaGo and Parallel Intelligence

Based on the ACP approach, the parallel intelligence can be defined as one form of intelligence that is generated from the interactions and executions between real and virtual systems. The parallel intelligence is characterized by being data-driven, artificial systems based modeling, and computational experiments based system behavior analytics and evaluation.

The core philosophy of parallel intelligence is, for a complex system, constructing a parallel system which is consist of real physical systems and artificial systems. The final goal

of parallel intelligence is to make decisions to drive the real system so that it tends to the virtual system. In this way, the complex system problems are simplified utilizing the virtual artificial system, and the management and control of the complex system are achieved.

Fig. 5 demonstrates a framework of a parallel system. In this framework, parallel intelligence can be used in three operation modes, i.e., 1) Learning and training: the parallel intelligence is used for establishing the virtual artificial system. In this mode, the artificial system might be very different from the real physical system, and not much interaction is required; 2) Experiment and evaluation: the parallel intelligence is used for generating and conducting computational experiments in this mode, for testing and evaluating various system scenarios and solutions. In this mode, the artificial system and the real system interact to determine the performance of a proposed policy; 3) Control and management: parallel execution plays a major role in this operation mode. The virtual artificial system and real physical system interact and refer between each other in parallel and in real-time, and thus achieve control and management of the complex system.
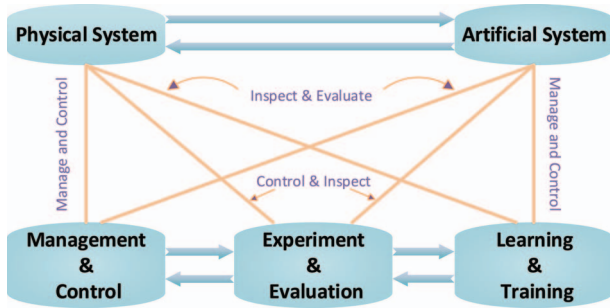

Fig. 5. The Virtual-Real Duality: The execution framework of the parallel systems.

We would like to point out that one single physical system can interact with multiple virtual artificial systems. For example, corresponding to different demands from different applications, one physical system can interact simultaneously or in a time-sharing manner with the data visualization artificial system, ideal artificial system, experimental artificial system, contingency artificial system, optimization artificial system, evaluation artificial system, training artificial system, learning artificial system, etc.

Using the framework of parallel systems, AlphaGo's parallel intelligence lies in the fact that 1) In the learning and training operation mode, AlphaGo is able to effectively learn from historical Go game data to establish its artificial system; 2) In the experiment and evaluation operation mode, AlphaGo is able to generate a huge amount of self-play games for reinforcement learning; and 3) In the control and management operation mode, Alphago is able to effectively search for decision strategies in real-time within its virtual system space, and interact with the physical system for making the optimal decision.

VII. Concluding Remarks

Recently, a Go match between an intelligent computer program, AlphaGo, and a top-tier human player, Lee Sedol, brought a tremendous and extensive impact to the field of AI. Due to the prohibitively high complexity of Go games and the need for fuzzy interpretation of the overall game situations, computer Go programs were considered to be incompetent in achieving intelligence in this field. However, AlphaGo made the breakthrough by defeating 4:1 a professional *nine dan* human player.

We discuss the potential prospects of the technology impacts brought by the victory of AlphaGo, rather than discussing its technical details. Obviously, the AlphaGo approach can be applied in many areas other than the Go game, and the further development of related AI approach may lead to the new intelligent technology (New IT) era. The victory of AlphaGo is also a persuasive augment for shifting the paradigm from scientific casuality to data association by computing in decision making, especially for complex systems.

A historical retrospective is provided on the important milestones which are related to the AlphaGo approach. From the retrospective, it is pointed out that the success of AlphaGo is necessity of fortuity, standing on the achievements of AI. For this reason, if we consider the AlphaGo within the context of AI history, one can provide a better vision in projecting the AlphaGo techniques into the future of AI.

Therefore, we propose a conjecture of the AlphaGo Thesis, which states an AlphaGo-like program is equivalent to a human-being with limited resources. Further, we propose another conjecture of the Extended AlphaGo Thesis, which states that a network-like system with sufficient number of adjustable parameters is equivalent to a human-being with limited resources. The network-like system, in the form of deep rule-based networks, is described in Section V utilizing Fuzzy Logic Rule sets. The DRN represents the design philosophy of "Simple Locally, Complex Remotely" with the implication of an implementable technical scheme.

Combined with the theoretical frameworks of the ACP approach and parallel systems, the implication of The AlphaGo Thesis is brought to the level of complex system intelligence. The ACP approach, aiming to mitigate the "paradox of scientific methods" in complexity systems' "scientific solutions", provides a triable and repeatable study methodology to complex systems, especially the ones involving human and societies. The ACP approach can be deemed as a generalized form of the AlphaGo approach, and it is described in details in Section VI. The "parallel intelligence" is directly derived and generated from applying the ACP approach in a complex system. With the strong evidence provided by AlphaGo's victory, together with the theoretical concepts such as software-defined systems, computational experiments and closed-loop feedback between parallel virtual-real systems, we foresee that parallel intelligence becomes a low-hanging fruit of AI in the near future.

Where does AlphaGo go? Where does the AlphaGo Thesis lead to? Where does the concept of parallel intelligence imply? The answers to these questions may need decades of efforts of AI researchers. However, it is foreseeable and unquestionable that many complex system problems in engineering management, economic management and society management are going to benefit from the generalized AlphaGO approach, the

ACP, and parallel system framework while working towards the answers of these questions.

ACKNOWLEDGMENT

REFERENCES

[1] Taves M. Google's AlphaGo isn't taking over the world, yet [Online], Available: http://www.cnet.com/news/googles-alphago-isnt-taking-over-the-world-yet/, March 13, 2016.

[2] Wang F Y. The Era of New IT, Human still cannot be replaced by machines [Online], Available: http://opinion.huanqiu.com/1152/2016-03/8689740.html, March 11, 2016.

[3] Tromp J. Number of legal Go positions [Online], Available: https://tromp.github.io/go/legal.html, 2016.

[4] Zobrist A L. Feature Extraction and Representation for Pattern Recognition and the Game of Go [Ph. D. dissertation], University of Wisconsin, USA, 1970

[5] Millen J K. Programming the game of Go. *Byte*, 1981, **6**(4): 102−114

[6] Green H S. Go and artificial intelligence. *Computer Game-Playing: Theory and Practice*. New York: Halsted Press, 1983. 141−151

[7] Webster B. A Go board for the macintosh. *Byte*, 1984, **9**(12): 125−128

[8] Silver D, Huang A, Maddison C J, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, **529**(7587): 484−489

[9] Wedd N. Human-Computer Go challenges [Online], Available: http://www.computer-go.info/h-c/index.html, 2011.

[10] Rimmel A, Teytaud O, Lee C S, Yen S J, Wang M H, Tsai S R. Current frontiers in computer Go. *IEEE Transactions on Computational Intelligence and AI in Games*, 2010, **2**(4): 229−238

[11] Tian Y D, Zhu Y. Better computer Go player with neural network and long-term prediction. arXiv: 1511.06410, 2016.

[12] Coulom R. Efficient selectivity and backup operators in Monte-Carlo tree search. In: Proceedings of the 5th International Conference on Computers and Games. Turin, Italy: Springer, 2007. 72−83

[13] Minsky M. *The Society of Mind*. New York: Simon & Schuster, 1988.

[14] Wang F Y, Saridis G N. Suboptimal control for nonlinear stochastic systems. In: Proceedings of the 31st IEEE Conference on Decision and Control. Tucson, AZ: IEEE, 1992. 1856−1861

[15] Wang F Y, Zhang H G, Liu D R. Adaptive dynamic programming: an introduction. *IEEE Computational Intelligence Magazine*, 2009, **4**(2): 39−47

[16] Wang F Y. Let's Go: From AlphaGo to parallel intelligence. *Science & Technology Review*, 2016, **34**(7): 72−74 (in Chinese)

[17] Wang F Y. Complexity and intelligence: From Church-Turing Thesis to AlphaGo Thesis and beyond (1). *Journal of Command and Control*, 2016, **2**(1): 1−4 (in Chinese)

[18] Conway F, Siegelman J. *Dark Hero of the Information Age: In Search of Norbert Wiener, the Father of Cybernetics*. New York: Basic Books, 2005.

[19] Burks A W, Wright J B. Theory of logical nets. *Proceedings of the IRE*, 1953, **41**(10): 1357−1365

[20] Shannon C E. Computers and automata. *Proceedings of the IRE*, 1953, **41**(10): 1234−1241

[21] Burks A W, McNaughton R, Pollmar C H, Warren D W, Wright J B. Complete decoding nets: general theory and minimality. *Journal of the Society for Industrial and Applied Mathematics*, 1954, **2**(4): 201−243

[22] Burks A W, Wang H. The logic of automata. *Journal of the ACM*, 1957, **4**(3): 279-297

[23] Rabin M O, Scott D. Finite automata and their decision problems. *IBM Journal of Research and Development*, 1959, **3**(2): 114−125

[24] Rabin M O. Complexity of computations. *ACM Turing Award Lectures*. New York, NY, USA: ACM, 2007.

[25] Cook S A. The complexity of theorem-proving procedures. In: Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, 1971, 151−158

[26] Wang F Y, Saridis G, McNaughton R. On Complexity of Intelligent Machines: an ε-complexity Approach. RPI RAL Report, 1988.

[27] Blum A L, Rivest R L. Training a 3-node neural network is NP-complete. *Neural Networks*, 1992, **5**(1): 117−127

[28] Budinich M. Neural networks for NP-complete problems. *Nonlinear Analysis: Theory, Methods & Applications*, 1997, **30**(3): 1617−1624

[29] Ghaziri H, Osman I H. A neural network algorithm for the traveling salesman problem with backhauls. *Computers and Industrial Engineering*, 2003, **44**(2): 267−281

[30] Potvin J Y. The traveling salesman problem: a neural network perspective. *ORSA Journal on Computing*, 1993, **5**(4): 328−348

[31] Yao X. Finding approximate solutions to NP-hard problems by neural networks is hard. *Information Processing Letters*, 1992, **41**(2): 93−98

[32] de Chardin P T. *The Phenomenon of Man*. London: William Collins Sons, 1955.

[33] Wang F Y, Kim H M. Implementing adaptive fuzzy logic controllers with neural networks: a design paradigm. *Journal of Intelligent & Fuzzy Systems*, 1995, **3**(2): 165−180

[34] Wang F Y. Parallel control: a method for data-driven and computational control. *Acta Automatica Sinica*, 2013, **39**(4): 293−302 (in Chinese)

[35] Wang F Y. Building knowledge structure in neural nets using fuzzy logic. in *Robotics and Manufacturing: Recent Trends in Research Education and Applications*, edited by M. Jamshidi, New York, NY, ASME (American Society of Mechanical Engineers) Press, 1992.

[36] Wang F Y. Hierarchical Structure and Linguistic Description for Complex Decision Networks: A Evaluation Programming Approach. PARCS Report, University of Arizona, Tucson, Arizona, USA, May 1999.

[37] Wang F Y. Parallel system methods for management and control of complex systems. *Control and Decision*, 2004, **19**(5): 485−489 (in Chinese)

Fei-Yue Wang received his Ph.D. in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, New York in 1990. He joined the University of Arizona in 1990 and became a Professor and Director of the Robotics and Automation Lab (RAL) and Program in Advanced Research for Complex Systems (PARCS). In 1999, he founded the Intelligent Control and Systems Engineering Center at the Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Oversea Chinese Talents Program from the State Planning Council and "100 Talent Program" from CAS, and in 2002, was appointed as the Director of the Key Lab of Complex Systems and Intelligence Science, CAS. In 2011, he became the State Specially Appointed Expert and the Director of the State Key Laboratory of Management and Control for Complex Systems.
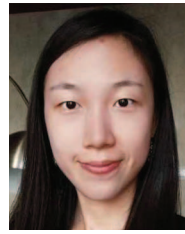
Dr. Wang's current research focuses on methods and applications for parallel systems, social computing, and knowledge automation. He was the Founding Editor-in-Chief of the *International Journal of Intelligent Control and Systems* from 1995 to 2000, and the EiC of *IEEE Intelligent Systems* from 2009 to 2012. Currently, he is the EiC of *IEEE Transactions on Intelligent Transportation Systems*. Since 1997, he has served as General or Program Chair of more than 20 IEEE, INFORMS, ACM, ASME conferences. He was the President of IEEE ITS Society from 2005 to 2007, Chinese Association for Science and Technology (CAST, USA) in 2005, and the American Zhu Kezhen Education Foundation from 2007-2008. Since 2008, he is the Vice President and Secretary General of Chinese Association of Automation.

Dr. Wang is elected Fellow of IEEE, INCOSE, IFAC, ASME, and AAAS. In 2007, he received the 2nd Class National Prize in Natural Sciences of China and awarded the Outstanding Scientist by ACM for his work in intelligent control and social computing. He received IEEE ITS Outstanding Application and Research Awards in 2009 and 2011, respectively. In 2014, Dr. Wang received the IEEE SMC Society Norbert Wiener Award.

Jun Jason Zhang (S'06–M'09–SM'14) received his B.E. and M.E. degrees in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2003 and 2005, respectively, and his Ph.D. in electrical engineering from Arizona State University, U.S.A., in 2008. He is currently an Assistant Professor of electrical and computer engineering at the University of Denver. His research interests are in the areas of agile sensing, signal processing and implementation, time-varying system modeling, and their applications in intelligent power and energy systems.
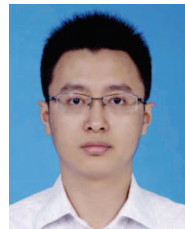
Xinhu Zheng received the B.S. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2011. He is currently working toward the Ph.D. degree in computer science and engineering at the University of Minnesota, Minneapolis, MN, USA. His research interests include social computing, machine learning, and data analytics.
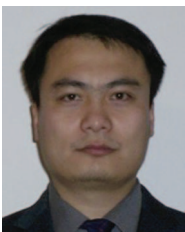
Xiao Wang received her Bachelor's degree in network engineering from Dalian University of Technology, in 2011. She is currently a Ph.D. candidate in the State Key lab of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. Her research interests include social transportation, cyber movement organizations, and social network analytics.

Yong Yuan received the B.S., M.S., and Ph.D. degrees in computer software and theory from the Shandong University of Science and Technology, Shandong, China, in 2001, 2004, and 2008, respectively. He is an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and also with the Qingdao Academy of Intelligent Industries, Qingdao, China. His current research interests include social computing and business intelligence.

Xiaoxiao Dai is currently a Ph.D. candidate in the Ritchie School of Engineering and Computer Science, University of Denver, Denver, Colorado, USA. She received her M.S. degree in Electrical Engineering from the Ritchie School of Engineering and Computer Science, University of Denver, Denver, Colorado, USA, in 2013, her B.E. degree in Electronics and Microelectronics from Jilin University, Changchun, China, and from National Research Tomsk Polytechnic University, Tomsk, Russia, in 2011. Her research interests include sensing, machine learning, signal processing, and network analysis.

Jie Zhang is an Assistant Professor in the State Key Lab of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences since 2011. His current research interests include mechanism design and optimal control in E-Commerce and transportation systems. He received his Ph.D degree in Technology of Computer Application from University of Chinese Academy of Sciences in 2015. He received his BSc. degree in Information and Computing Science from Tsinghua University in 2005, and received MSc. degree in Operations Research and Control Theory from Renmin University of China in 2009.

Liuqing Yang (S'02–M'04–SM'06–F'14) received the Ph.D. degree from University of Minnesota, Minneapolis, MN, USA, in 2004. She is currently a Professor at the Colorado State University, Fort Collins, CO, USA. Her general interests are in signal processing with various applications, on which she has published more than 235 journal and conference papers. Dr. Yang has been actively serving the technical community. She has been involved in the organization of many IEEE international conferences, and has served on the Editorial Board for a number of journals, including *IEEE Transactions on Signal Processing*, *IEEE Transactions on Communications*, *IEEE Transactions on Wireless Communications*, and *IEEE Transactions on Intelligent Transportation Systems*. She received the ONR YIP Award in 2007, the NSF CAREER Award in 2009, the IEEE GLOBECOM Outstanding Service Award in 2010, and Best Paper Awards at IEEE ICUWB'06, ICCC'13, ITSC'14, and Globecom'14.