Abhi Mangal Agarwal

Analysis of Go due to the emergence of computers, information systems and artificial intelligence in the 20th and 21st century.

In this paper I will be analyzing the game of Go both through a perspective of developing the game by using artificial intelligence and computer systems and also the impact that computers and information systems have had on the game since they have been developed to play locally and through a wide area network on a computer.

History of Go and Computers

Firstly, I will be discussing the general history of Go and computers to put this vastly researched academic topic into perspective. Programming in Go began in late 1960s and was primarily done by people who played go and academic researchers in fields of Computer Science and Information Systems (Burmeister). The aim was not to contribute to the programming theory of Go directly, but rather to use Go to further their other research aims and use the concepts of Go in other areas of development. Serious research in developing Go programs advanced after the spread of personal computers and when the financial prizes like the Ing prize were announced.

The earliest Go program was written by Albert Zobrist as a part of his PhD thesis on recognizing patterns. He introduced the idea of "using an influence function to segment the board into black and white territories" (Burmeister). His idea was to primarily give a value of +50 to white stones and -50 to black stones, and empty points would be valued at 0. Neighboring same colored stones would give a positive influence value of +1 and different colored stones would give a negative value of -1. This process was repeated 4 times in total, and the aim was to spread black and white influence numerically. Furthermore, Jon Ryder came out with a solution that was historically seen as an extension to Zobrist's method; his program used the same concept of using an influence function but he was able to find a different solution to the problem. He saw the game as one which required a balance between actual territory and potential territory and also as a game where one avoids the loss of important stones to the opponent.

After Zobrist, Walter Reitman and Bruce Wilcox began to conduct research in Go in 1972. They were able to produce a program called INTERIM.2, which won against ranked 22 and 34 kyu and one lost against a player ranked 4 kyu. The computer rank for the program was 27 kyu, which was a breakthrough in that day and age. There was a long line of people who tried to produce a Go program, from Cosmos in 1990, to Zhixing Chenin 2000 who wrote a 4 kyu ranked program.

Nowadays there is a greater incentive to produce these games, as the Ing Chang-ki Wei-Chi Educational Foundation gives out an Ing prize worth of 1.5 million dollars for programs globally, to beat a nominated human competitor without handicap on a 19x19 board.

Initial problems

As history suggests it's not easy to produce a game of this caliber like Go. But this section will be discussing the initial problems that occurred when developing and engineering Go on the computer.

Programming in Go was initially not well developed because there was a strong commitment to research on programming Chess, which didn't happen with the Go field. The performance of Go programs have not kept up with the performance of Chess programs due to the fundamental differences between these games, which make it harder to develop a computerized version of a game like Go. These difficulties have been faced when creating a high level of game play, such as when playing with even a 1d level player. I have included some of the difficulties which are experienced by Go programmers:

- Even though there are fewer types of pieces in Go than in Chess, the board size of Go is much larger than in Chess (8x8 in Chess compared with typically 19x19 in Go).

- The size of the board and freedom of choice of putting the stones at different positions is relatively higher than in chess, averaging 80! moves in Chess vs. 361! (*Coates*) moves in Go.
- In Chess there are a number of well-known openings the computer could recognize, but in Go the number of openings is larger (hence calculating the probability is harder), and research shows that the set openings in Go are usually not followed after the first three moves (*Coates*).
- In Go it is difficult for the computer to recognize when the game is over, and for beginners it is usually over when both players pass, unlike Chess where most players can tell when a checkmate occurs. The natural end occurs playing additional pieces will reduce the players score, either by filling in their own territory or, giving unnecessary prisoners to the opponent. In both of these cases the Go system would be inefficient and it would take a large amount of time to calculate each move just because it would spend time checking for these endings.
- Each stone in Go can have long term effects on in the intensity and outcome of the game. Even though a stone in Go does not move after it's placed, a group of stones can play a big role in a capturing race or can affect the life and death struggle of another group. For example, a stone played in the path of a ladder can change the outcome of the game and the alter moves that could have been thought out previously. This exponentially increases the number of calculations that the computer has to compute in each move in order to determine the best path.

Considering all these examples, the system we have to design and consider in this case would have to be powerful enough to run through millions of calculations to find the probabilistic outcome of moves, like it does in chess, but also run calculations to try to realize that the game is coming to an end and that it should pass in order to not decrease its score. The ever-changing nature of the game also makes it difficult not just for a computer, but also a human to figure and map out the game.

Apart from the issues in being able to program a core artificial intelligence component a quote from the *New Scientist* by I.J. Good verifies another problem that is still applicable even though it was initially written in 1965:

"Go on a computer? – In order to programme a computer to play a reasonable game of Go, rather than merely a legal game – it is necessary to formalise the principles of good strategy, or to design a learning programme. The principles are more qualitative and mysterious than in chess, and depend more on judgment. So I think it will be even more difficult to programme a computer to play a reasonable game of Go than of chess."

This is a particularly important problem that is difficult to be combatted by engineers and computer scientists in designing the game. Through machine-learning aspects, they have been able to design algorithms that have been able to teach the A.I component when it has not encountered a certain problem. However, it is very difficult for it to play with the 6d level as it wouldn't be able to visualize the logic of certain moves, as it would lack the principles of good strategy and would lack intuition to a certain level. It would be trying to learn a particular move in a mathematical way and would not be able to relate it to the philosophical or deep level of cultural understanding behind it. *The Art of Warfare* shows that there are certain characteristics in war and in Go that can be explained through very philosophical and historical methods and this is something difficult to have explained or programmed into a bot.

For the reasons listed above about programming approaches to chess, some of these approaches are manageable in using binary tree algorithms but such approaches have not succeeded in Go. The factor of probabilities or possibilities in Go is too large for such algorithms. The game of Go is constantly changing in probability as stones which could have initially been linked together, can be captured and the opponent can drastically change his position in the game at any point in comparison to chess, where there are a computable number of outcomes as the game advances.

Emergence of Artificial Intelligence

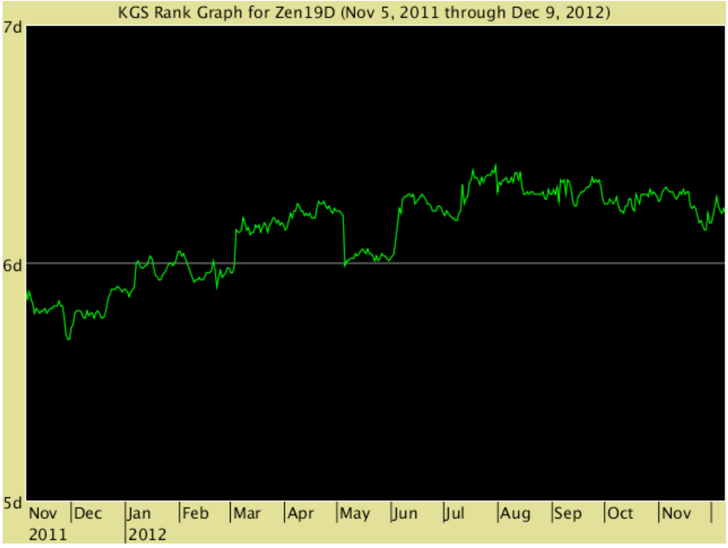
The emergence and development of artificial intelligence is a key step that has also helped the development of Go. Currently, Go is a topic that is widely debated and tackled within A.I. and there is also a field in A.I. dedicated to creating a computer program that plays Go.

Moreover, some forms of artificial intelligence are essentially programming themselves and cannot always explain the moves the other players make. A computer program can be a combination of syntax and semantics, and data and behavior of the program depends heavily on the data. Artificial intelligence consists of a mathematical model of neural networks (*Coates*). A neural network is used to model the complex relationships between inputs and outputs to find patterns and to learn from the given data, and in this case, the scenarios that will be posed during a Go game. This gives the bot the ability to judge and evaluate the scenario, and random disturbances of the data will lead to a better performance as the neural net learns to play out the different scenarios.

Success

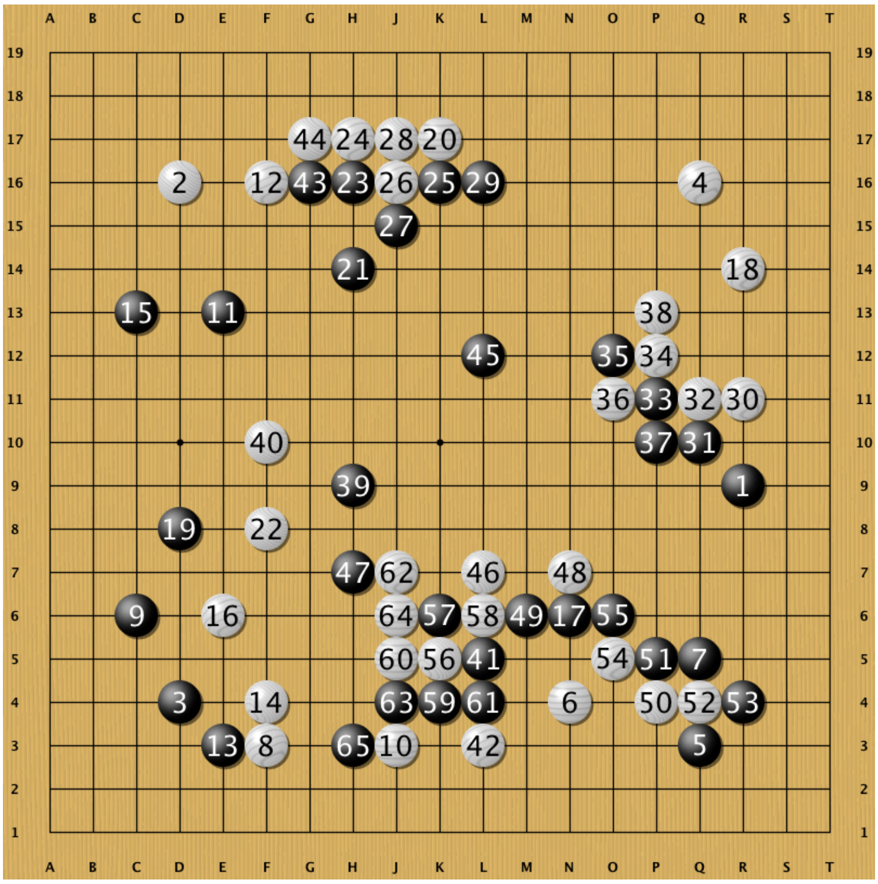
In recent times they have been able to develop a system called Zen19D (*KGS*), which is a version of Zen19; it runs on a very powerful machine and is a product that is being academically developed by team DeepZen. Zen as a category of bots reached 1d level 2 years ago, and has been making ground-breaking progress in the field moving to 5d level players in 2011.

It is estimated to be around 5d for shorter games and 4d for longer games, and throughout the year it is able to defeat 3d level players even if they take handicap and is now considered a solid 5d level player. It won the 2009 Computer Olympiad and here is a graph of the KGS server in which it operates in and Zen was the first bot to hold a KGS 3d rating for more than 20 rated games in a row (*Ball*).



The graph (*KGS)* shows that the machine is able to learn some of the moves itself that the other player uses, and that it uses machine learning to understand the steps that are played out. Then in other games, it is able to recognize these patterns and apply the same mathematical probability using its neural networks. The bot is currently able to play with anyone on the KGS server and it's always busy playing with someone and therefore it's able to learn new moves and steps it hasn't encountered every day and is always improving its level. It is predicted through a vast number of polls that if it improves at the same rate as it currently is it will be able to reach professional dan level within one year.

One of the most amazing successes of the Go computer programs was the games played with John Tromp (*Cook*) a well known 2 dan level player (EGF rankings). In 1997 John Tromp made a bet that the computer would not be able to beat him (in 2010), and he was right- he was able to beat the computer 4-0. But the game changer was in 2012 when the computer was able to beat him 3-1 and the development from losing by 4 points to winning by 2 points shows the progress made by the Go programming community in developing these systems. The games were very interesting, especially the third game, and the computer was able to use very tactical and impressive moves against the player. Some of the plays, which will be displayed below are some of the moments where it seemed like the bot was thinking like a human:



The Black was Zen19N and White was John Tromp, and after 65 moves Tromp had resigned the game giving Black an automatic victory. It was interesting to see that the bot was targeting the center from a very early stage (moves 1, 11, 15, 21, 27, 45) and targeting White very closely, pushing the opponent towards the wall (moves 43, 23, 25, 27, 29). It seemed like a very mathematical way to approach it to push White towards the wall and gain the center, but it also entails a very humanistic approach using intuition send in a way self-defense. To protect the stones, the machine pushes him towards the outside and gaining the center. The machine also was able to attack aggressively towards the white stones (moves 5, 7, 51, 53) and contain those stones in that particular position.

On the other hand, people were also commenting on how Black had an advantage because it was a bot and would not tire even after 2 hours into the game, which is something that needs to be considered, but the player should know what he is getting into before he plays the game with a machine. They felt that it was unfair for the machine to have such a big advantage, and looking at time they had remaining the machine had more than double the time left in comparison.

Future of Computer Go

In an interview conducted by David Ormerod (*Ormerod*) with Martin Muller, a researcher in Go and automated planning, he explained that the Monte Carlo algorithm is the way the advanced Go systems in the future systems will be designed. The algorithm uses "statistics collected from randomized simulations as a way to analyze complex systems which are too hard to 'solve' by other means" (*Ormerod*). It is suggested that for the opening, some of the machine programs have used pattern recognition and match them with the popular openings by human experts, and therefore they had to be hardcoded into the system rather than the machine learning it through experience. The Monte Carlo algorithm will patch the weaknesses and will focus on "probably good" (*Ormerod*) moves, and also be able to more efficiently learn and filter through the openings without them being hardcoded. There are some particular moves in Go where the current machine solutions aren't efficient, examples including seki (mutual life), semeai (capturing race), ko fights, and nakade (move inside).

Moreover, as processing power also improves and increases the computer will be able to run faster processors and therefore there is a coloration between the processing power and the number of simulations. It has been researched by Don Dailey in 2008 (*Dailey*) and has come out to be a logarithmic curve[1].

His basic argument was that the improvement in Go has a logarithmic relationship between the number of processors and the number of simulations the computer is able to run, and therefore this improvement in speed will lead to the computer being able to calculate probabilities faster. Moreover, Muller also suggest the addition of a third dimension to the game and local search where people can play out simulations within the game, and local search which would be able to keep track of simultaneous fights. This will allow people to in a way play out and search through their moves, which doesn't particularly seem like in the spirit of the game but would be a technological break-through.

Impact of the development of Go on the Computer

Technology has started to make an impact in all areas of board games, and in the experience of playing an "analog game" (*Case*). It is inevitable that a strong social hobby, even board games, would be affected by the emergence of the Internet and ways to digitalize the game. We are drawn towards the digital versions of these games as producers strive to make it as realistic as possible, and people are fascinated by both the special effects and the simplicity of it.

This is something that is also happening to the game of Go, although not to the same extent as games like Monopoly, but people are trying to digitalize the game to allow play against each other without being in the same location. This has its advantages and its disadvantages; the key advantage

[1] http://cgos.boardspace.net/study/smoothscale.jpg

would be that it does allow people to play with people who they have never met in person. However, it also loses its beauty and the natural experience of putting down the stone. The current generation prefer to play Go on the computer in their leisure time, rather than practicing on the board, as the repeatability and convenience of putting and moving through the plays is simpler, and is much more preferred. In addition, people are also able to find players who are willing to teach them or are of their same level to help them in developing in the game, and the Internet or World Wide Web allows them to access all these resources.

The development of artificial intelligence components into these games is interesting as well. People are now able to play against computers if they have no one to play against, and these computers can go from beginner level to professional dan level nowadays and they are available at any point on the Go servers online. It is interesting to see the development of Go as they are able to play 6d level players and defeat them, which would be incredible in the computer science society as the field of artificial intelligence considers Go as a better measure of a computer’s capacity for thought and progress than chess. The emergence of more programs to play online at every ranking level allows people to train efficiently and quickly with players, the programs, who maintain and can play through the whole game with the same high level of gameplay. This helps individuals who are not able to find challengers at their level or people who are playing in their spare time.

In conclusion, the development of Go in general in the field of artificial intelligence and information systems is rapid and will constantly be evolving at least throughout the next decade. However, the emergence of the game online and played through digital means poses two questions: would the emergence of computers decrease the number of players who play the game on the board in the coming generations? Or would it do the opposite and attract more people because it’s freely available? These questions are difficult to answer and looking back at chess it has become freely available and has helped Chess grow and created a huge network of chess players online. Moreover, go development has also lead to further advancements in pattern recognition, game theory, and many other fields and further advancements in go (due to the prize) will also compliment a lot of other technologies.

Bibliography

Darren Cook. (2009). *The Shodan Go Bet.* Available: http://dcook.org/gobet/. Last accessed 8th December 2012.

Jay Burmeister . (1995). *CS-TR-339 Computer Go Tech Report.* Available: http://itee.uq.edu.au/~janetw/Computer%20Go/CS-TR-339.html#3.0. Last accessed 9th December 2012.

KGS. (2010). *Bot beats 6-dan amateur (even game).* Available: http://gosensations.com/?id=2&server_id=1&new_id=1158. Last accessed 9th December 2012.

KGS. (2010). *Games of KGS player zen19, December 2012.* Available: http://www.gokgs.com/gameArchives.jsp?user=zen19. Last accessed 9th December 2012.

Elwyn Berlekamp. (). *Mathematical Go: Chilling Gets the Last Point.* Available: http://math.berkeley.edu/~berlek/cgt/gobook.html. Last accessed 5th December 2012.

Feng - Hsiung Hsu. (October 2007). *Cracking GO.* Available: http://spectrum.ieee.org/computing/software/cracking-go. Last accessed 6th December 2012.

Andy Coates. (). *Go Artificial Intelligence.* Available: http://ai-depot.com/LogicGames/Go.html. Last accessed 7th December 2012.

Abhi Mangal Agarwal

Chris Ball. (2012). *Computers are very good at the game of Go.* Available: http://blog.printf.net/articles/2012/02/23/computers-are-very-good-at-the-game-of-go/. Last accessed 9th December 2012.

Richard Bozulich. (). *The Magic of Go.* Available: http://gostart.ru/e011.html. Last accessed 8th December 2012.

Quaestor. (2012). *MoGo.* Available: http://senseis.xmp.net/?MoGo. Last accessed 9th December 2012.

Loyd Case. (2011). *Living with Technology: Technology's Impact on Board Games.* Available: http://www.tested.com/tech/gaming/3015-living-with-technology-technologys-impact-on-board-games/. Last accessed 10th December 2012.

David Ormerod. (2012). *An interview with Martin Müller.* Available: http://gogameguru.com/computer-go-demystified-interview-martin-mueller/. Last accessed 10th December 2012.

Don Dailey. (2008). *9x9 Scalability study.* Available: http://cgos.boardspace.net/study/index.html. Last accessed 10th December 2012.