The game of Go may have simple rules, but to play successfully requires enormous skill. Few computer programs are able to navigate through the unimaginable number of possible moves and find the best strategy. The solution, as shown by MoGo, is not even to attempt to look at all the possibilities. Instead, a randomised exploration of the options based on multi-arm bandits makes this problem more manageable. The secret behind the early success of MoGo was Upper Confidence Trees.

The idea of using machines to take the role of our opponent in a game arose even before computers had been created. Chess has long attracted the attention of designers of these machines. While the very earliest chess automata were frauds, by the 1950s the subject was being considered seriously. According to Claude Shannon in 1950, we should think of the game as an imaginary tree branching out from the current state of the board. Each possible move was another branch from the current state; follow a branch and you reach a new board state from which new branches grow. Chess has a lot pieces and a lot of possible moves by each piece, making the game tree vast. Even the most powerful supercomputers cannot search all of this tree. However by the late 1990s and early 2000s, our best computers were fast enough to search significant parts of the tree. Computers such as Deep Blue and later Deep Fritz were able to beat human chess grand masters using these brute-force approaches.

But the game of Go was another matter entirely. Compared to chess, which may have around 37 legal moves possible each turn on average, Go has rarely fewer than 50 moves possible, and typically over 200 each turn. This means that the game tree for Go is considerably larger than the one for chess. So even with the very fastest supercomputers, a brute-force search for good moves simply doesn't work for the game of Go.

> This project (or adventure...) has been really a wonderful collaborative work which has led to some advances not only in the field of Go, but also which has opened new perspectives in other fields as well.

Monte Carlo search is one useful way to make vast search spaces more manageable. Instead of trying to exhaustively search the whole space, a Monte Carlo method randomly samples points in the space and then uses the results of that sampling to approximate the solution. It's like the game of Battleship, where random sampling is used to build an approximate picture of where the ships are on the grid. When applying Monte Carlo search to a game tree, this equates to evaluating many branches of the tree chosen randomly to see

which player might eventually win or lose if those moves were followed, and then deciding on the best branch to follow. Traditional Monte Carlo methods might iteratively search the game tree, deeper and deeper, but they sampled each branch with a uniform probability. An even better approach, as introduced by researchers Levente Kocsis and Csaba Szepesvári in 2006, is to bias the random sampling of branches. Using a method called Upper Confidence Trees (UCT) (an extension of a previous method known as Upper Confidence Bounds) the different moves within the game tree

are treated like different one-armed-bandit machines. Each time a one-armed-bandit is played it will return a random result, but on average some provide better results than others. In exactly the same way, each time a certain move is tried, it might result in a different outcome (depending on all the other moves), but on average some are better than others. By finding an appropriate balance between exploring which are the best bandits (moves) and exploiting the good ones already found, the Monte Carlo method can be biased towards more useful moves in the game tree. In this way UCT is able to search the game tree of Go selectively and randomly and discover a good move to make each time. (In reality “multi-arm bandits” are more appropriate for Go than one-arm bandits.)

In 2006 a group of researchers took exactly this approach when they developed their Computer Go playing program. Prof Remi Munos and his student Yizao Wang (a keen player of Go) at Ecole Polytechnique were inspired by the recent work of Rémi Coulom who had created a hugely successful Monte Carlo Go player called Crazy Stone. They joined forces with another student, Pierre-Arnaud Coquelin, and Olivier Teytaud and his PhD student Sylvain Gelly at University of Paris-Sud. This team applied the UCB algorithm to the game tree of Go and without realising it, independently invented the UCT algorithm at almost the same time as Kocsis and Szepesvári.

MoGo used UCT to bias its Monte Carlo search – a strategy that enabled the program to win the CGOS tournament where games were played on a smaller 9x9 grid, in August 2006. MoGo was first introduced in the On-line Trading of Exploration and Exploitation Workshop 2006 (sponsored by PASCAL, the Network of Excellence on Pattern Analysis, Statistical Modelling and Computational Learning).

The same team went on to use a Monte Carlo and bandit-based approach and win the 2006 “PASCAL Exploration vs. Exploitation Challenge,” set by TouchClarity (now Omniture TouchClarity), see box.

Despite the successes of the MoGo team, other researchers remained sceptical about their approach. According to Sylvain Gelly, “at that time, the ideas we were using in MoGo were considered ‘interesting’ in the Computer Go community only for the 9x9 Go, but were considered doomed by the fact that they were not successful on the "real" game. 9x9 Go was considered to be another game unrelated to 19x19 Go.”

But the MoGo team did not lose hope and Sylvain Gelly spent the next year improving MoGo (with methods such as Rapid Action Value Estimation). By March 2007 MoGo won a 19x19 Computer Go tournament against strong classical 19x19 programs, and was honourably ranked in 19x19 games against humans, which attracted interest outside the Computer Go community for these new techniques. MoGo continued to defy expectations, as Gelly explained: “Interestingly, the improvements made for the 19x19 board turned out to be useful for the 9x9 board as well (while people were claiming that the two games were unrelated), and MoGo in the mean time achieved "master" level (amateur) in 9x9, which also attracted interest outside Computer Go.”

Before long, MoGo was beating all other programs it played in the Kiseido Go Server tournaments, for several months in a row. Since then, MoGo has had many success, including winning the Gold Medal in the 2007 Computer Game Olympiad for the game of Go, and being the first program to defeat a professional player in 9x9 Go. In March 2008 MoGo won a 9x9 game against the winner of the "Tournoi de Paris", the most important tournament in Europe, and Catalin Taranu (fifth Dan Pro) claimed that mogo was close to the Dan level in 19x19.

The idea of bandit algorithms used recursively for performing efficient tree search is a new research direction that is definitely worth investigating both from theoretical and practical perspectives.

According to Olivier Teytaud, one of the current developers of MoGo, “MoGo does not contain a lot of Go expertise and is however quite strong; this shows that the approach is quite general... We are now applying UCT and related tools to other games and to some combinatorial optimisation problems.”

From game-playing to webpage optimisation, Monte Carlo bandit methods seem to provide a powerful new technique. In the words of Remi Munos, “I think the idea of bandit algorithms used recursively for performing efficient tree search (such as UCT) is a new research direction that is definitely worth investigating both from theoretical and practical perspectives.”

TouchClarity was recently acquired by Omniture, specialists in online business optimisation. The development of MoGo has also been a fruitful avenue of research for all involved. In 2007 Yizao Wang and Sylvain Gelly won the prize for best student paper in the IEEE Symposium on Computational Intelligence and Games. Wang also received an honor "prix d'option" from Ecole Polytechnique for his work on this project. Gelly was awarded his PhD on “A Contribution to Reinforcement Learning; Application to Computer Go” in September 2007 and now works at Google, Zurich. Coulom continues to improve Crazy Stone

Omniture TouchClarity focuses on optimising the online experience for users – it decides what should and should not be shown within webpages, based on the interaction of past users. The technology is being used by most UK Banks (and many other clients) to provided tailored promotions to visitors based on their clicks within the pages. The problem can be regarded as being similar to a game where the human players (users) make their moves by clicking on certain parts of web pages, and the computer responds with its move of changing the content to make it more likely that the user will choose options of interest to the users (and profitable for the company). So the same kinds of technology used for playing Go can also enable this problem to be solved. Chief Scientist Leonard Newnham explained, “the PASCAL challenge allowed us to revisit the problem from the ground up… discussions with PASCAL members helped us to refine our algorithms.”