AI techniques for the game of Go

Summary

The thesis describes our research results and the development of new AI techniques that improve the strength of Go programs. In chapter 1, we provide some general background information and introduce the topics of the thesis. We focus on two important lines of research that have proved their value in domains which are related to and relevant for the domain of computer Go. These lines are: (1) searching techniques, which have been successful in games such as Chess, and (2) learning techniques, which have been successful in other games such as Backgammon, and in other complex domains such as image recognition. For both lines we investigate the question to what extent these techniques can be used in computer Go.

Chapter 2 introduces the reader to the game of Go. Go is the most complex popular board game in the class of two-player zero-sum perfect-information games. It is played regularly by millions of players in many countries around the world. Despite several decades of AI research, and a million-dollar prize for the first computer program to defeat a professional Go player, there are still no Go programs that can challenge a strong human player.

Chapter 3 introduces the standard searching techniques used in this thesis. We discuss minimax, $\alpha\beta$, pruning, move ordering, iterative deepening, transposition tables, enhanced transposition cut-offs, null windows, and principal variation search. The searching techniques are applied in two domains with a reduced complexity to be discussed in chapters 4 and 5.

Chapter 4 investigates searching techniques for the task of solving the capture game, a simplified version of Go aimed at capturing stones, on small boards. The main result is that our program PONNUKI solved the capture game on empty square boards up to size 5×5 (a win for the first player in 19 plies). The 6×6 board is solved, too (a win for the first player in 31 plies), under the assumption that the first four moves are played in the centre. These results were obtained by a combination of standard searching techniques, some standard enhancements adapted to exploit domain-specific properties of the game, and a novel evaluation function. We conclude that standard searching techniques and enhancements can be applied effectively for the capture game, especially when they are restricted to small regions of fewer than 30 empty intersections. Moreover, we conclude that our evaluation function performs adequately at least for the task of capturing stones.

Chapter 5 extends the scope of our searching techniques to Go, and applies them to solve the game on small boards. We describe our program MIGOS in detail. It uses principal variation search with (1) a combination of standard search enhancements (transposition tables, enhanced transposition cut-offs), improved search enhancements (history heuristic, killer moves, sibling promotion), and new search enhancements (internal unconditional bounds, symmetry lookups), (2) a dedicated heuristic evaluation function, and (3) a method for static recognition of unconditional territory. In 2002, MIGOS was the first Go program in the world to have solved Go on the 5×5 board.

We analyse the application of the situational-super-ko rule (SSK), identifying several problems that can occur when the history of a position is discarded by the transposition table, and investigate some possible solutions. Most problems can be overcome by only allowing transpositions that are found at the same depth in the search tree. The remaining problems are quite rare, especially in combination with a decent move ordering, and can often be ignored safely.

We conclude that, on current hardware, provably correct solutions can be obtained within a reasonable time frame for confined regions of a size up to about 28 intersections. For efficiency of the search, provably correct domain-specific knowledge is essential to obtain tight bounds on the score early in the search tree. Without such domain-specific knowledge, detecting final positions by search alone becomes unreasonably expensive.

Chapter 6 introduces the learning techniques used in this thesis. We explain the purpose of learning, and give a brief overview of the learning techniques that can be used for game-playing programs. Our focus is on multi-layer perceptron (MLP) networks. We discuss the strengths and weaknesses of the possible network architectures, representations and learning paradigms, and test our ideas on the simplified domain of connectedness between stones. We find that training and applying complex recurrent network architectures with reinforcement learning is quite slow, and that simpler network architectures may provide a good alternative especially when large amounts of training examples and well-chosen representations are available. Consequently, in the following chapters we focus on supervised learning techniques for training simple architectures to evaluate moves and positions.

Chapter 7 presents techniques for learning to predict strong moves from game records. We introduce a training algorithm which is more efficient than standard fixed-target implementations due to the avoidance of needless weight adaptation when predictions are correct. As an extra bonus, the algorithm reduces the number of gradient calculations as the performance grows, thus speeding up the training. A major contribution to the performance is the use of feature-extraction methods. Feature extraction reduces the training time while increasing the quality of the predictor. Together with a sensible scaling of the original features and an optional second-phase training, superior performance over direct-training schemes can be obtained.

The predictor can be used for move ordering and forward pruning in a full-board search. The performance obtained on ranking professional moves indicates that a large fraction of the legal moves may be pruned directly. Ex-

periments with the program GNU Go indicate that a relatively small set of high-ranked moves is sufficient to play a strong game against other programs.

Our conclusion is that it is possible to train a learning system to predict good moves most of the time with a performance at least comparable to strong kyu-level players. Such a performance can be obtained from a simple set of locally computable features, thus ignoring a significant amount of information which can be obtained by more extensive (full-board) analysis or by specific goal-directed searches. Consequently, there is still significant room for improving the performance further.

Chapter 8 presents learning techniques for scoring final positions. We describe a cascaded scoring architecture (CSA*) that learns to score final positions from labelled examples, and apply it to create a reliable collection of 9×9 game records (which is re-used for the experiments in chapters 9 and 10). On unseen game records CSA* scores around 98.9% of the positions correctly without any human intervention. By comparing numeric scores and counting unsettled interior points nearly all incorrectly scored final positions can be detected (for verification by a human operator). Although some final positions are assessed incorrectly by CSA*, it turns out that many are in fact scored incorrectly by the players. Detecting games that were incorrectly scored by the players is important for obtaining reliable training data.

We conclude that for the task of scoring final positions supervised learning techniques can provide a performance at least comparable to reasonably strong kyu-level players. This performance is obtained by a cascade of relatively simple classifiers in combination with a well-chosen representation, which only employs features that are calculated statically (without search).

Chapter 9 focuses on predicting life and death. The techniques from chapter 8 are extended to train MLP classifiers to predict life and death in non-final positions too. Experiments show that, averaged over the whole game, around 88% of all blocks (that are relevant for scoring) are classified correctly. Ten moves before the end of the game 95% of all blocks are classified correctly, and for final positions over 99% are classified correctly. We conclude that supervised learning techniques in combination with a well-chosen representation can be applied quite well for the task of predicting life and death in non-final positions. At least for positions near the end of the game we are confident that the performance is comparable to that of reasonably strong kyu-level players.

Chapter 10 investigates various learning techniques for estimating potential territory. Several direct and trainable methods for estimating potential territory are discussed. We test the performance of the direct methods, known from the literature, which do not require an explicit notion of life and death. Additionally, two enhancements for adding knowledge of life and death and an extension of Bouzy's method are presented. The experiments show that without explicit knowledge of life and death the best direct method is Bouzy's method extended with a means to divide the remaining empty intersections. When information about life and death is used to remove dead stones, the difference with distance-based control and influence-based control becomes small, and it is seen that all three methods perform quite well.

The trainable methods are new and can be used as an extension of the system for predicting life and death presented in chapter 9. A simple representation for our trainable methods suffices to estimate potential territory at a level outperforming the best direct methods. Experiments show that all methods are greatly improved by adding knowledge of life and death, which leads us to conclude that good predictions of life and death are the most important ingredient for an adequate full-board evaluation function.

Here we conclude that supervised learning techniques can be applied quite well for the task of estimating potential territory. When provided with sufficient training examples, these techniques easily outperform the best direct methods known from literature. On a human scale, we are confident that for positions near the end of the game the performance is at least comparable to that of reasonably strong kyu-level players. However, without additional experiments it is difficult to say whether the performance is similar in positions that are far away from the end of the game.

Finally, chapter 11 revisits the research questions, summarises the main conclusions, and provides directions for future research. Our conclusion is that domain-specific knowledge is the most important ingredient for improving the strength of Go programs. For small problems sufficient provably correct knowledge can be implemented by human experts. When searches are confined to regions of about 20 to 30 intersections, the current state-of-the-art searching techniques together with adequate domain-specific knowledge representations can provide strong and often even perfect play. Larger problems require heuristic knowledge. Although heuristic knowledge can be implemented by human experts, too, this tends to become quite difficult when the playing strength of the program increases. To overcome this problem learning techniques can be used to extract automatically and intelligently knowledge from the game records of human experts. For maximising performance, the main task of the programmer then becomes providing the learning algorithms with adequate representations and large amounts of reliable training data. When both are available the quality of the static knowledge that can be obtained using learning techniques is at least comparable to that of reasonably strong kyu-level players. The various techniques presented in this thesis have been implemented in the Go program MAGOG which enabled it to win the bronze medal in the 9×9 Go tournament of the 2004 Computer Olympiad in Ramat-Gan, Israel.

Samenvatting

Dit proefschrift beschrijft onderzoek naar en de ontwikkeling van nieuwe AI-technieken om de speelsterkte van Go-programma's te verbeteren. Hoofdstuk 1 geeft enige algemene achtergrondinformatie en introduceert de onderwerpen van dit proefschrift. We richten ons op twee belangrijke onderzoekslijnen die hun waarde hebben bewezen in domeinen die verwant zijn met het domein van computer-Go. Het zijn: (1) zoektechnieken, die succesvol zijn geweest in spelen zoals schaken, en (2) leertechnieken, die succesvol zijn geweest in andere spelen zoals Backgammon, en in andere complexe domeinen zoals beeldherkenning. Voor beide onderzoekslijnen beschouwen we de vraag in hoeverre deze technieken bruikbaar zijn in computer-Go.

Hoofdstuk 2 introduceert het spel Go. Go is het meest complexe populaire bordspel in de klasse van tweepersoons nulsom spelen met volledige informatie. Wereldwijd zijn er miljoenen mensen die regelmatig Go spelen. Ondanks tientallen jaren van AI-onderzoek, en een prijs van zo'n 1 miljoen dollar voor het eerste programma dat een professionele speler zou verslaan, maken Go-programma's nog altijd geen schijn van kans tegen sterke amateurs.

Hoofdstuk 3 bevat een overzicht van de standaard zoektechnieken die in dit proefschrift worden gebruikt. We behandelen minimax, $\alpha\beta$, snoeien, het ordenen van zetten, iteratief verdiepen, transpositie-tabellen, *enhanced transposition cut-offs*, *null windows*, en *principal variation search*. De zoektechnieken worden toegepast in twee domeinen met een gereduceerde complexiteit. Zij worden behandeld in de hoofdstukken 4 en 5.

Hoofdstuk 4 richt zich op zoektechnieken voor het oplossen van Slag-Go (een vereenvoudigde versie van Go gericht op het slaan van stenen) op kleine borden. Het belangrijkste resultaat is dat ons programma Ponnuki Slag-Go heeft opgelost voor kleine lege borden tot de grootte van 5×5 (de eerste speler wint na 19 zetten). Het 6×6 bord is ook opgelost onder de aanname dat de eerste vier zetten in het centrum worden gespeeld (de eerste speler wint na 31 zetten). Deze resultaten zijn verkregen met behulp van (1) standaard zoektechnieken, (2) enkele standaard verbeteringen waarbij gebruik gemaakt is van domeinspecifieke eigenschappen van het spel, en (3) een nieuwe evaluatiefunctie. We concluderen dat standaard zoektechnieken en de verbeteringen effectief toegepast kunnen worden voor het spel Slag-Go, met name als het domein beperkt is tot kleine gebieden met minder dan 30 lege kruispunten. Verder concluderen we dat de evaluatiefunctie in ieder geval geschikt is voor het vangen van stenen.

In hoofdstuk 5 breiden wij het domein van de zoektechnieken uit van Slag-Go naar Go, en passen ze toe bij het oplossen van Go op kleine borden. We beschrijven het programma MIGOS in detail. Het maakt gebruik van *principal variation search* met een combinatie van standaard verbeteringen (transpositie-tabellen, *enhanced transposition cut-offs*), aangepaste verbeteringen (*history heuristic, killer moves, sibling promotion*), en nieuwe verbeteringen (*internal unconditional bounds, symmetry lookups*), een heuristische evaluatiefunctie, en een methode voor het statisch herkennen van onconditioneel gebied. In 2002 heeft MIGOS als eerste programma ter wereld Go op het 5×5 bord opgelost.

Voorts analyseren we in dit hoofdstuk de toepassing van de situationele-super-ko regel (SSK). We beschrijven verscheidene problemen die zich kunnen voordoen in combinatie met de transpositie-tabel als de geschiedenis van een positie wordt genegeerd, en onderzoeken enkele mogelijke oplossingen. De meeste problemen zijn oplosbaar door alleen transposities te gebruiken die op dezelfde diepte in de zoekboom gevonden zijn. De resterende problemen zijn zeldzaam, vooral in combinatie met een degelijke zettenordening, en kunnen meestal veilig genegeerd worden.

We concluderen dat, op hedendaagse hardware, bewijsbaar correcte oplossingen verkregen kunnen worden binnen een redelijke tijd voor afgesloten gebieden met maximaal zo'n 28 kruispunten. Voor efficiënt zoeken is bewijsbaar correcte domeinspecifieke kennis essentieel voor het verkrijgen van stringente grenzen aan de mogelijke scores in de zoekboom. Zonder gebruik te maken van zulke domeinspecifieke kennis is het herkennen van eindposities, puur op basis van zoeken, in de praktijk niet goed mogelijk.

Hoofdstuk 6 geeft een overzicht van de leertechnieken die in dit proefschrift gebruikt worden. We leggen het doel van het leren uit, en geven aan welke leertechnieken gebruikt kunnen worden. Onze aandacht is gericht op meerlaags perceptron (MLP) netwerken. We bespreken de sterke en zwakke kanten van mogelijke architecturen, representaties en leerparadigma's, en testen onze ideeën op het vereenvoudigde domein van connectiviteit tussen stenen. Onze bevindingen laten zien dat het trainen en toepassen van complexe recurrente netwerk-architecturen met *reinforcement learning* bijzonder traag is, en dat eenvoudiger architecturen een goed alternatief zijn, vooral als grote aantallen leervoorbeelden en goed gekozen representaties beschikbaar zijn. Derhalve richten wij ons in de hoofdstukken 7, 8, 9 en 10 op *supervised* leertechnieken voor het trainen van eenvoudige architecturen voor het evalueren van zetten en posities.

Hoofdstuk 7 presenteert technieken voor het leren voorspellen van sterke zetten op basis van opgeslagen partijen. We introduceren een trainingsalgoritme dat efficiënter is dan standaard implementaties omdat het geen onnodige berekeningen uitvoert wanneer de ordeningen correct zijn en daarmee het aantal noodzakelijke gradiënt-berekeningen reduceert, wat de training aanzienlijk versnelt naarmate de voorspellingen beter worden. Een belangrijke bijdrage aan de prestatie is het gebruik van kenmerk-extractie. Kenmerk-extractie reduceert de leertijd en verbetert de kwaliteit van de voorspellingen. In combinatie met een zinnige schaling van de originele kenmerken en een tweede-fase training kunnen superieure prestaties worden verkregen ten opzichte van directe training.

De voorspellingen kunnen gebruikt worden om zetten te ordenen en voorwaarts te snoeien bij het zoeken op het volledige bord. De kwaliteit van de ordening van professionele zetten geeft aan dat een groot deel van de legale zetten direct gesnoeid kan worden. Experimenten met het programma GNU Go laten zien dat een relatief klein aantal hoog geordende zetten voldoende is om een sterke partij te spelen tegen andere programma's.

Onze conclusie is dat het mogelijk is om een lerend systeem te trainen dat zetten kan voorspellen op een niveau dat minstens vergelijkbaar is met dat van sterke kyu-spelers. Dit niveau is reeds haalbaar op basis van relatief eenvoudige lokaal te berekenen kenmerken, waarmee we dus een aanzienlijke hoeveelheid informatie negeren die verkregen kan worden door middel van een uitgebreidere analyse (van het volledige bord) of door middel van het speciaal doelgericht zoeken. Derhalve zijn er nog voldoende verbeteringen mogelijk.

Hoofdstuk 8 presenteert leertechnieken voor het waarderen van eindposities. We beschrijven een cascade-scoring architecture (CSA*) die leert om eindposities te waarderen op basis van gelabelde voorbeelden. We passen deze methode toe om een betrouwbare verzameling 9×9 partijen samen te stellen (die tevens gebruikt wordt voor de experimenten in hoofdstuk 9 en 10). Op onafhankelijke partijen scoort CSA* volledig zelfstandig zo'n 98,9% van de posities correct. Door numerieke scores te vergelijken en onbesliste interne punten te tellen kunnen bijna alle incorrect gescoorde eindposities gedetecteerd worden (voor verificatie door een menselijke operator). Hoewel sommige eindposities incorrect beoordeeld worden door CSA*, blijkt dat het merendeel incorrect beoordeeld is door de spelers. Het herkennen van partijen die incorrect beoordeeld zijn door de spelers is belangrijk voor het verkrijgen van betrouwbaar leermateriaal.

We concluderen dat *supervised* leertechnieken voor het waarderen van eindposities een prestatieniveau halen dat zeker vergelijkbaar is met redelijk sterke kyu-spelers. Dit niveau wordt verkregen met behulp van relatief eenvoudige beslissers in combinatie met een goed gekozen representatie, die slechts gebruik maakt van kenmerken die statisch berekend kunnen worden (zonder zoeken).

Hoofdstuk 9 richt zich op het voorspellen van leven en dood. De technieken uit hoofdstuk 8 worden uitgebreid voor het trainen van MLP-beslissers om leven en dood te voorspellen in niet-eindposities, ook op basis van gelabelde voorbeelden. De experimenten laten zien dat, gemiddeld over de gehele partij, zo'n 88% van alle blokken (die relevant zijn voor de score) correct geclassificeerd worden. Tien zetten voor het einde van de partij wordt zo'n 95% van alle blokken correct geclassificeerd, en voor de eindposities wordt meer dan 99% correct geclassificeerd. We concluderen dat *supervised* leertechnieken in combinatie met een adequate representatie behoorlijk goed in staat zijn om ook in niet-eindposities leven en dood te voorspellen. We zijn er van overtuigd dat, in ieder geval voor posities vlak voor het einde van de partij, het prestatieniveau vergelijkbaar is met dat van redelijk sterke kyu-spelers.

Hoofdstuk 10 onderzoekt verscheidene leertechnieken voor het schatten van potentieel gebied. Verschillende directe en lerende methoden voor het schatten van potentieel gebied worden behandeld. We testen de prestaties van de directe methoden die bekend zijn uit de literatuur en geen expliciete notie van leven

en dood nodig hebben. Tevens presenteren we twee verbeteringen voor het toevoegen van kennis omtrent leven en dood en een uitbreiding van Bouzy's methode. De experimenten laten zien dat zonder expliciete kennis van leven en dood Bouzy's methode, indien die uitgebreid is met een methode om de neutrale punten verder te verdelen, de beste directe methode is. Als informatie over leven en dood gebruikt wordt om de dode stenen te verwijderen, is het verschil met *distance-based control* en *influence-based control* klein, en doen alle drie de methoden het behoorlijk goed.

De lerende methoden zijn nieuw en kunnen gebruikt worden als uitbreiding van het systeem voor het voorspellen van leven en dood dat we beschreven hebben in hoofdstuk 9. Een eenvoudige representatie voldoet om potentieel gebied beter te schatten dan de beste directe methoden. Experimenten laten zien dat alle methoden aanzienlijk beter presteren met kennis van leven en dood. Dit leidt tot de conclusie dat kennis van leven en dood het belangrijkste ingrediënt is van een adequate evaluatiefunctie voor het gehele bord.

Hier concluderen wij dat *supervised* leertechnieken behoorlijk goed toegepast kunnen worden voor de taak van het schatten van potentieel gebied. Wanneer voldoende trainingsvoorbeelden beschikbaar zijn kunnen deze technieken de beste directe methoden uit de literatuur eenvoudig verslaan. Op de menselijke schaal zijn we ervan overtuigd dat vlak voor het einde van de partij de prestaties minstens vergelijkbaar zijn met die van redelijk sterke kyu-spelers. Echter, zonder aanvullende experimenten is het lastig om te zeggen of dat ver voor het einde ook zo is.

Tenslotte komen we in hoofdstuk 11 terug op de onderzoeksvragen, geven een overzicht van de belangrijkste conclusies, en eindigen met suggesties voor nader onderzoek. Onze conclusie is dat domeinspecifieke kennis het belangrijkste ingrediënt is voor het verbeteren van de sterkte van Go programma's. Voor kleine problemen kan voldoende bewijsbaar correcte kennis geïmplementeerd worden door menselijke experts. Als het zoeken begrensd is tot gebieden van ongeveer 20 tot 30 kruispunten zijn de huidige *state-of-the-art* zoektechnieken in combinatie met adequate domeinspecifieke kennisrepresentaties in staat tot zeer sterk en vaak zelfs perfect spel. Voor het aanpakken van grotere problemen is heuristische kennis noodzakelijk. Hoewel heuristische kennis ook geïmplementeerd kan worden door menselijke experts wordt dit naarmate het programma sterker wordt snel lastiger. Dit laatste kan vermeden worden door gebruik te maken van leertechnieken die automatisch en op intelligente wijze kennis extraheren uit de partijen van menselijke experts. Voor het maximaliseren van de prestaties wordt de belangrijkste taak van de programmeur dan het leveren van adequate representaties en grote hoeveelheden betrouwbare leervoorbeelden aan het leeralgoritme. Als beide beschikbaar zijn is de kwaliteit van de statische kennis die kan worden verkregen met leertechnieken minstens vergelijkbaar met die van spelers van redelijk sterk kyu-niveau. De technieken die we in dit proefschrift hebben gepresenteerd zijn geïmplementeerd in het Go-programma MAGOG en hebben direct bijgedragen aan het winnen van de bronzen medaille in het 9×9 Go-toernooi van de 9de Computer Olympiade (2004) in Ramat-Gan, Israel.