THOMAS WOLF [1]

A Dynamical Systems Approach for Static Evaluation in Go

[1]Department of Mathematics, Brock University, St.Catharines, Canada and ZIB Fellow

A Dynamical Systems Approach for Static Evaluation in Go

Thomas Wolf

Brock University

St. Catharines, Ontario, Canada

WWW: http://www.brocku.ca/mathematics/people/wolf/

Email: twolf@brocku.ca

Abstract

In the paper arguments are given why the concept of static evaluation (SE)[1] has the potential to be a useful extension to Monte Carlo tree search. A new concept of modeling SE through a dynamical system is introduced and strengths and weaknesses are discussed. The general suitability of this approach is demonstrated.

I. MOTIVATION

The concept of Monte-Carlo simulations applied to go [1] combined with the UCT algorithm [2], [3], which is a tree search method based on Upper Confidence Bounds (UCB) (see e.g. [4]) produced a new type of programs like [5], [6], [7] that dominate computer go in recent years. The detailed tournament report [8] of the program MoGo playing against professional and amateur players reveals strengths and weaknesses of MoGo which are typical for programs that perform a Monte Carlo Tree Search (MCTS).

With the significant progress especially in 9×9 go where MoGo running on large hardware reached professional level, it needs good reasons to start work on a static evaluation function (SE) in go. The following are indications that, although being a large step forward, MCTS will not be able to take programs to professional playing level purely on its own:

- The ratio log(increase in needed computing power) / (increase in strength) is too big to get to professional strength with pure MCTS in the foreseeable future.
- Compared with human playing strength the playing level of MCTS decreases with increasing board size from 9x9 to 13x13 to 19x19.
- There are still children under 10 years old with a clearly higher playing strength than the best programs running on the largest clusters available. Even if computing power could be increased by currently unimaginable amounts and minors could be beaten in the game, it would be unsatisfying, if playing strength would have to rely so massively on hardware.

One way to equip MCTS with more go knowledge is to add it directly by having heuristic procedures to detect special situations and add a bonus to the number of simulations won by moves that it recommends.

The problem with this approach seems to be that

- either the situations that allow precise recommendations of moves are too special and thus happen too rarely, so that the constant effort invested in recognizing such positions is too high compared with the improvement MCTS would gain given the same extra time, or
- the situations are more general, but this generality prevents an accurate recommendation of the best move or a safe recommendation of which moves to ignore, or
- anything in between.

What is needed is an approach that has an understanding of the local and global situation in *general* positions, that either is accurate to some extent or indicates in which areas it is not accurate. On top of that it needs to be fast. What simplifies matters is that it does not have to be perfect in any respect. Lack of accuracy could be compensated with a higher speed, or it could be slow if its accuracy matches the quality of the intuition of a strong human player.

The SE proposed in this paper is applicable to any position, it is reasonably fast and has in general a good understanding of the situation. Before supporting this statement with test details in section V, we argue in the following section why a high quality SE is not unrealistic and how it should be structured, discuss limits of the design in section III and give details of the dynamical system in section IV.

In section VII we comment on necessary future improvements of the static evaluation. The appendix gives an example how an influence function can guide finding the best move which MCTS would have to accomplish through more expensive global search.

---

[1] Among users of the Internet Go server KGS the abbreviation SE is used for 'Score Estimator'. Although different from 'Static Evaluation' a score estimator is easily obtained from static evaluation by adding up probabilities of chains to be alive at the end of the game or points to be owned by White or Black.

II. Initial Considerations

A. Resources unused in pure MCTS

Why should it be possible to design a static evaluation which can provide at least some information faster and/or better than MCTS?

One strength of MCTS is to be useful for programs playing other games than go or even for optimization tasks not involving games. This strength is at the same time a weakness when applied to go: MCTS does not take advantage of simplifying aspects of the nature of go:

1) *Chains are local.* Chains connect adjacent stones of the same colour into a unit, so that all of them are captured, or none of them are. Stones that are not connected to the chain do not belong to it.
2) *Capturing a chain is local.* To capture a chain the opponent must fill all of the chain's adjacent intersections. Opposing stones placed further away do not capture the chain.
3) *Go has an influence field.* Although at the end of the game it is clear for each point who owns it and for each chain whether it is alive or dead, so the state of each point and chain is fully described by 1 bit, this is not the case earlier in the game.

   At earlier stages it makes sense to introduce a field which could be called 'influence' or 'strength' which is useful as a model to guide towards optimal play, not only for humans.

   This influence field is not simply a tool to accommodate human slowness in reading compared to MCTS. The point to make is that this influence-field is in some sense real and can be characterized and modeled. It shows, for example, some stability or null-sum property and can explain higher playing level sacrifice moves as done with a position in the appendix.

   We want to state it as a conjecture: *To maximize playing strength for a given amount of computational power (cycles per sec and given amount of memory, both sufficiently large but fixed) a field embodying strength, influence and perhaps other fields / state variables have to be introduced.* This is not different from progress in the natural sciences and mathematics where the improvement of quantitative knowledge and an evolution of the scientific language depend on each other.

   For human go-players the importance of knowing about influence is self-evident. The point to make is that MCTS misses out on the guiding potential of a sufficiently accurate influence function. MCTS gets its strength from statistical learning but its efficiency depends on the size of the search space which increases exponentially with the board size. The cost of computing a good static evaluation does not increase exponentially with the board size.

4) *Influence varies smoothly.* The influence of stones falls off smoothly, at least in the opening in non-life-and-death situations. Also, the example in the appendix shows the need of intermediate influence values other than 1 (full domination) and 0 (no influence at all). More discussion on smoothness in go can be found in section II.A. in [9].

B. Design Decisions

Based on the above observations the following design decisions have been made for the static evaluation function.

- The strength values of chains and influence values at points are represented by floating point numbers because of points 3) and 4) above but also to evaluate some fuzzy knowledge by a number that changes smoothly with the degree of certainty of the knowledge.
- Because of 1), 2) the set of all relations of neighbouring points and chains is formulated as a single dynamical system of algebraic relations expressing the strength of each chain and influence at a point in terms of the strength of neighbouring chains and influence at neighbouring points. A strength value is assigned to a whole chain, irrespective of its size or shape. All that matters in this approximation are the neighbourhood relations. A **s**tatic **e**valuation based on **d**ynamical **s**ystems will be abbreviated as SEDS in the remainder of the paper.

A different question is whether the SE should depend on who moves next. Although it may become slightly better by taking that into account (e.g. if two important chains of opposite colour touch each other and have only one liberty each, so that the side to move next may capture the opponent's chain), the SE to be described does not use who moves next. Reasons are:

- It is not obvious how to use who moves next without prejudice even in the simple case of, say, Black moving next and white chains under atari being so small that their capture has low priority. Another example is the case when many white chains are under atari.
- Making the SE dependent on who moves next is not a general solution. It may take 2 moves to simplify the all-or-nothing fight so that the SE can 'see' the outcome, or 3, 4, ... moves. The issue of merging SE and MCTS has to be solved more rigorously, not by a quick fix of making SE dependent on who moves next.
- The value of moving next naturally varies from area to area. To consider it properly would imply to know the value for each area but that essentially means to be able to play perfectly. This would be contradictory to the philosophy of splitting up the problem of determining the best move into three parts: designing a static evaluation, a search procedure (MCTS) and an interplay between both.

III. Limits of what Dynamical Systems can do

Not all rules of go are local by nature. The rule that players alternate in their moves sets limits to the usability of SEDS which are to be discussed in this section. On the other hand, in a local fight both sides may not alternate their moves. If the fight does not have highest priority then one side may not answer an opponents move and play elsewhere. A different example of non-alternating moves is given in the appendix where a sacrifice allows Black to move twice in a row in a crucial area.

A. Ladders

The action at a distance inflicted by ladder breaking stones seems to be a good example against local models like our dynamical system model. But as with plane waves in physics with long distance effect being described through *local* differential equations one can not easily exclude that ladder breaking stones could be described through a local model. However, ladders are a good counterexample against static evaluation. For example, to work out a long winding ladder like in diagram 1 statically - even if possible at all in principle - would be so much more difficult than simply performing the moves in a deep but narrow tree search.
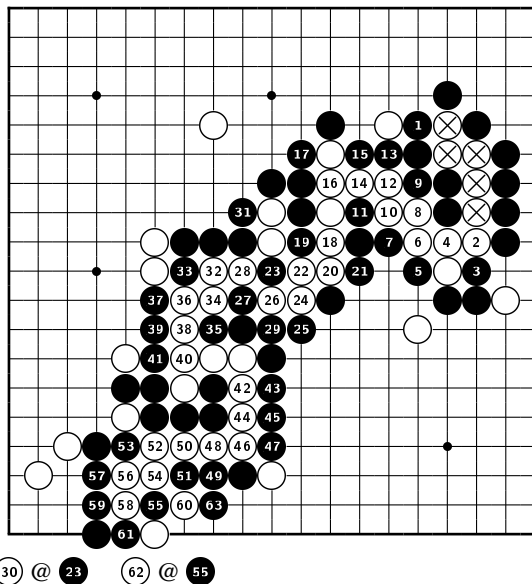

Diagram 1. ● catches ⊗ in a winding ladder.

B. Life & Death

An example for a concept in go that is *not* a purely local phenomenon, i.e. it can not be described by considering only one chain/point and its neighbours at a time, is the concept of *life* which is defined recursively: *A chain is alive if it participates in at least two living eyes and an eye is alive if it is surrounded only by living chains.* To identify unconditional life one has to consider the complete group of living chains at once.
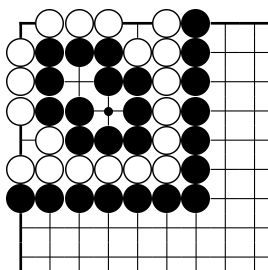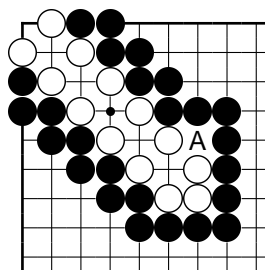

Diagram 2.   Diagram 3.

For example, the lives of the white chains in diagram 2 depend on each other and the conclusion that all chains are alive can only be drawn at once, not in an iterative way and not by considering one chain and its neighbours at a time, so not by a purely local algorithm. Similarly, the life of the white stones in diagram 3 hangs on who moves next at the *distant* point A in a *discrete, non-iterative* way.

The current version of a SEDS computer program recognizes (non-local) static life (life without ever having to answer any threat) at the time when neighbourhood relations are established during the initialization of SEDS. Although this is a first step towards including life and death in SEDS, static life happens only rarely in games.

IV. A DYNAMICAL SYSTEMS APPROACH

In this section we describe a conceptually simple dynamical systems model that was implemented and studied for its strengths and weaknesses.

A. The Setup

The elementary objects on the board (we call them units from now on) are taken to be all empty points and chains (for which no shape is recorded). Individual stones of a chain have no own identity in this model.

Based on the capture rule of go, units have completely local relations with each other, i.e. the state variables describing each unit can be computed explicitly from the state variables of neighbouring units and the resulting dynamical system can be solved iteratively.

This system couples all units on the board (i.e. all (empty) points and chains) and thus a fixed point of the dynamical system is a global consequence of the whole board. A change in strength of one chain would influence the strength of weak neighbouring chains and so on but the influence would stop at strong chains.

With points on the edge of the board having only 3 neighbours and in the corners only having 2 neighbours, the influence of the edge should come out properly without the need of extra artificial adjustments.

B. State Variables

To each *point* i (i.e. each empty intersection) are attached 2 real floating point type numbers:

w_i ... probability to be occupied by $\bigcirc$ at the end of the game
b_i ... probability to be occupied by $\bullet$ at the end of the game

and to each *chain* j is attached one number:

s_j ... probability for this chain to survive.

All values are in the interval $0 \ldots 1$.
For explanation purposes we also introduce

$\bar{w}_i, \ \bar{b}_i \ldots$ probability that at least one neighbouring point is occupied by resp. $\bigcirc$ or $\bullet$ at the end of the game.

C. The Relations

Apart from the trivial requirement that probabilities add up to 1:

$$b_i + w_i = 1 \tag{1}$$

the only assumption we make is $w_i/b_i = \bar{w}_i/\bar{b}_i$, i.e.

$$w_i \bar{b}_i = b_i \bar{w}_i. \tag{2}$$

At least in the extreme cases $(\bar{w}_i, \bar{b}_i) = (1,1), \ (1,0), \ (0,1)$ this relation is correct.

From (1), (2) we get

$$w_i = \frac{\bar{w}_i}{\bar{b}_i}(1 - w_i) = \frac{\bar{w}_i}{\bar{b}_i} - \frac{\bar{w}_i}{\bar{b}_i}w_i$$

$$w_i\left(1 + \frac{\bar{w}_i}{\bar{b}_i}\right) = \frac{\bar{w}_i}{\bar{b}_i}$$

$$w_i = \frac{\bar{w}_i}{\bar{b}_i}\left(1 + \frac{\bar{w}_i}{\bar{b}_i}\right)^{-1} = \frac{\bar{w}_i}{\bar{w}_i + \bar{b}_i} \tag{3}$$

where $\bar{w}_i, \bar{b}_i$ have to be expressed in terms of b_j, w_j, s_j from the neighbouring points and chains.

D. An Example Computation

In this simple example we are going to use relation (3) to compute the probability of points 1,2,3 in diagram 4 to be occupied finally by Black or White. For the simplicity of this example, the chains are set to be alive: $s_j = 1$. In the real model their

Diagram 4.

strengths would also be computed iteratively based on the strength of direct neighbouring chains and the influence of their direct neighbouring points.

To apply 3 we set $\bar{b}_1 = s(\text{leftblackstone}) = 1$ and $\bar{w}_1 = s(\text{whitestones}) = 1$ and get

$$\rightarrow \ w_1 = \frac{1}{1+1} = \frac{1}{2} = b_1 = w_3 = b_3 \quad (\text{by symmetry})$$

$$
\begin{aligned}
w_2 &= \frac{1}{1+\bar{b}_2} &\rightarrow \bar{b}_2 &= \text{probability of } \bullet \text{ on 1 or 3} \\
&= \frac{1}{1+3/4} & &= 1 - \text{probability of } \bigcirc \text{ on 1 or 3} \\
&= \frac{4}{7} & &= 1 - w_1 w_3 \\
& & &= 1 - \frac{1}{4} \\
b_2 &= \frac{3}{7} & \searrow &= \frac{3}{4}
\end{aligned}
$$

A similar computation is done for all chains where the probability of being captured is computed as the probability of all neighbouring points being occupied by the opponent and all attached opponent chains being alive.

This small example demonstrates how the computation goes but it also shows the limited value of the numbers obtained. They make sense if the moves are played randomly. In the derivation of the formulas all moves are assumed to be uncorrelated, but that is not the case: if White plays on 1 then Black plays on 3 and vice versa.

Another systematic error is made in relation 1. There are points where $w_i = b_i = 0$. For example, in a living white eye there is $w_i = 1$ although White will not play there but that is no problem if points with $w_i = 1$ are regarded as either occupied in future *or* owned by White. But points where relation 1 is truly violated are liberties that are shared by two chains in seki. Here is $w_i = b_i = 0$. The problem is not the territorial count with both points getting $w_i = b_i = 0.5$ but the safety of the chains. The SEDS can conclude from $w_i = b_i = 0$ that both chains are safe, but not from $w_i = b_i = 0.5$.

E. A Full Board Example
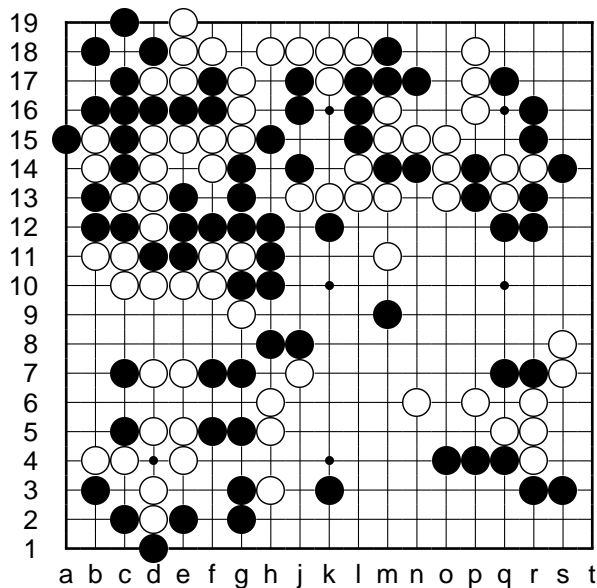
Before the iteration all b_i and w_i variables are initialized to 0.5 and all s_i variables are initialized to 1.0 .

In total there are 489 variables and as many equations. The following are just three of them:

$$
\begin{aligned}
w_{r8} &= (b_{q8}b_{r9}s_{r7}s_{s7} - b_{q8}b_{r9}s_{r7} + 1)/ \\
&\quad (b_{q8}b_{r9}s_{r7}s_{s7} - b_{q8}b_{r9}s_{r7} + \\
&\quad s_{r7}s_{s7}w_{q8}w_{r9} - s_{s7}w_{q8}w_{r9} + 2), \\
b_{r8} &= -w_{r8} + 1, \\
s_{r7} &= -s_{r4}s_{s7}w_{p7}w_{q6}w_{q8}w_{r8} + 1.
\end{aligned}
$$

The full set is shown on `http://lie.math.brocku.ca/ twolf/papers/WoSE2010/1`. Through this system each dynamical variable (2 for each point, 1 for each chain) is expressed in terms of the variables describing their neighbouring points and chains.

The system is iterated until all values change less than some threshold parameter.

V. RESULTS

A. Existence, Uniqueness and Stability

The dynamical systems formulated along the lines of the previous section have always at least two solutions: one solution where all white chains live, all black are dead and all empty points are fully under white influence and the same with switched colours. If all w_i, b_i, s_i are initialized according to one of these solutions, the iteration will keep these values stable.

In addition to these solutions in any board position computed so far the dynamical system had another solution (i.e. a fixed point) with all values in the interval 0..1. This solution was obtained from any initial conditions other than the ones leading to the two extreme solutions mentioned above.

A good question is how many iterations are necessary for all values to settle down so that all changes are less than some threshold parameter, say 10^{-5}. The interesting result is that for clear cut situations only few iterations (< 10) are necessary whereas for very unstable situations (for example, two attached chains both under atari) the number of iterations can reach hundreds or thousands. This opens the possibility to get as a by-product a measure of instability. On the other hand this requires more iterations than needed for a strength estimate.

B. Statistics on Professional Games

The SEDS function has been tested in detail by trying to predict the next move in professional games. Alternatively one can also look at these tests as tries to exclude as many moves as possible but the professional move if the main intention is to use the SEDS to narrow the search of MCTS.

The test consisted of doing a 1-ply search for all positions occurring in all 50,000 professional games from the GoGoD collection [10]. For each board position in these games this includes

- performing each legal move in this position,
- iterating the dynamical system in the new position until the system stabilizes,
- adding up all probabilities of chains to survive and points to be owned by either side and thus reaching a total score,
- ordering all legal moves according to their total scores,
- finding and recording the position of the professional move in the ranking of all moves.

The statistics have been recorded separately for each move number because at different stages of the game the static evaluation has different strengths and weaknesses.

The results of this test are reported under `http://lie. math.brocku.ca/twolf/papers/WoSE2010/2` due to the size of the diagram files. The data have been produced by evaluating 5.4 million positions from the 50,000 professional games of the GoGoD collection ([10]). On this web site three sequences of diagrams are shown. Each sequence contains over 400 diagrams, one for each move number. Figure 1 is one of the diagrams of the first sequence. It shows the number of positions in which the professional move lands in the range $x \ldots (x + 1)\%$ of legal moves as sorted by the SE where $x = 99$ are the top moves falling in the $99 \ldots 100\%$ range and $x = 0$ are the worst moves. Thus the higher the graph is on the right and the lower it is on the left, the better is the static evaluation.

Fig. 1. A statistics of the ranking of the next professional move according to SEDS in all positions with move number 50 from 50000 professional games

The second series of diagrams differs from the first by having a logarithmic vertical axis. This is useful if the emphasis is to safely ignore moves from further consideration in MCTS because then we want to be reasonably sure that SE does not rank good moves (moves played by the professional player) as bad (on the left side of figure 1). In other words, we want to be sure that the graph is low on the left and to highlight that range a logarithmic vertical scale is useful. Figure 2 is the logarithmic version of figure 1).

If one normalizes the vertical axis in figure 1 then this curve is the probability density $P(x)$ of the ranking of the professional move among all moves. If one accumulates this density from the right one obtains a so called 'survival function' $R(x)$: $R(x) := \int_x^{99} P(u)du$ which is displayed in figure 3. For example, a point on the graph with horizontal coordinate 85 and vertical coordinate 62 means: The professional move is kept with a probability of 62% (it survives) if the worst 85% of the moves are dropped (worst according to the SEDS).

Fig. 3.   The data of figure 1 here in a cumulative form of a survival function

C. Interpretation

In view of the simplicity of the static evaluation function SEDS the results as shown in figure 1 are surprisingly good.

Deficiencies are not difficult to explain. Because of the local concept of the dynamical system approach the SEDS has no concept of life (except a hard-wired fast recognition of static life), i.e. it does not know of the need of two eyes and the benefit of destroying eyes. For the SEDS in its current form (Dec 2009), strength is 100% correlated with resistance against being captured. As a consequence sacrifice moves, like Black on A in figure 4 get a low ranking. This is an extreme example where the professional move (Black on A) gets the lowest ranking of all moves by SEDS. Moves of this type make up the most left hump in figure 2.
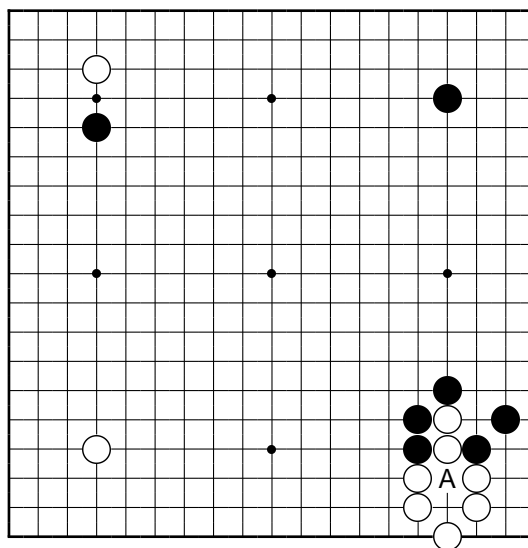

Fig. 4.   A position from a professional game where SEDS fails due to lacking a concept of eyes.

Whereas the humps on the left of figure 2 are due to good (professional) moves getting a low evaluation by SEDS, the dents on the right of the graph are due to bad moves getting a high evaluation by SEDS. The most right dent in figure 2 is due to the feature of the evaluation function to favour moves on the 2nd line, especially the 2-2 points. Again, this is a consequence of not knowing about the need for 2 eyes due to not knowing that Black and White can not do 2 moves at once, i.e. fill 2 eyes at once. For the SEDS a move on a 3rd or 4th line can be cut under by the opponent on the 2nd line because SEDS does not know that the move on the 2nd line needs two eyes.

When skipping through diagrams on `http://lie.math. brocku.ca/twolf/papers/WoSE2010/2` one sees that the current version of SEDS is most useful early in the game when life & death fights do not play a big role yet but also after about 15 moves when the exact influence of the edge of the board is not so crucial anymore.

Based on these findings it is expected that an appropriate consideration of life based on 2 eyes when computing the strength of chains will lead to a significant improvement of SEDS. The problem is to find a natural merger of the need for 2 eyes with the current interaction formula 3. Also, the concept of life is not local, so the solution of the dynamical system and the determination of a (non-local) measure of life based on 2 eyes must be merged naturally into one algorithm. Of course, one could make quick progress with a superficial repair but the aim of this exercise is to get a lasting concept that has no artificial parameters and no artificial constructs and thus scales (can improve indefinitely with increasingly available computer power) and thus has the potential to result in a strong program in the long run.

D. Timing

The following times have been recorded on a Dell Optiplex GX620 PC with Intel(R) Pentium(R) D CPU 3.40GHz processor with 2MB cache size running Linux. One CPU was used. Times reported in table 1 are the result of averaging the times for evaluating 400 positions from 400 professional games from the GoGoD game collection [10] each position on the same move number. The computations include making a move and updating the strength of chains and influence at points on the whole board.

move number	time in μs for static evaluation	time in ms for ranking moves
10	46.2	16.2
30	51.8	17.1
100	73.8	19.2
130	83.9	19.3
200	104.4	16.7
300	180.0	10.8

Table 1. Average times for static evaluation and ranking of moves.

As the update of the strength of a chain is slower to compute than an update of the influence at a point, the static evaluation (column 2) becomes slower the more stones are on the board. On the other hand, the more stones are on the board, the fewer legal moves exist and ranking all moves (column 3) becomes faster.

VI. SUITABILITY AND AVAILABILITY

For SEDS to be usable in its current form, the quality of the ranking it produces shown in figure 1 should be higher than the quality of a ranking MCTS is able to produce in the same time (column 3 of table 1). This has not been tested.

Independent of the outcome of such an experiment, SEDS in its current form has much potential for improvement. Also, it would become much more usable if one could use the results of a static evaluation directly in MCTS without performing a 1-ply search like now.

A run-time library under Linux is available that provides influence values at points and strength values of chains. It also has a function which sorts all legal moves by their estimated quality. To try it out within a MCTS program please contact the author.

VII. FUTURE TASKS

The dynamical system as formulated in section IV is a first version that allows us to study general properties of such an approach.

The following are possibilities to improve SEDS.

- The formula for the strength of a chain could be improved by giving the simple number of liberties a higher weight in the current probability computation.
- SEDS should provide a local awareness of stability (i.e. the dependence of the local strength measures on who moves next), of the size of an unstable area and thus of the importance of performing moves in this area,
- A recognition of safe links and simple life based on influence should be included in SEDS. The aim should be to improve the performance in guessing professional moves, more accurately, in excluding as many as possible moves other than the next moves in professional games.

Whereas MCTS is self-sufficient, SE is not. The merging of both will be a main future task. In doing this one would want to be able to vary smoothly the times allocated to both, at first statically then dynamically depending on the board position: more time for SE early in the game and in non-fighting positions, less time towards the end of the game and in all-or-nothing fights.

APPENDIX

In this appendix an example is given to support the claim that in go there is a field that embodies strength and influence, which is not an artificial human construct but which is at the heart of the game and is of intrinsic, fundamental nature. In the

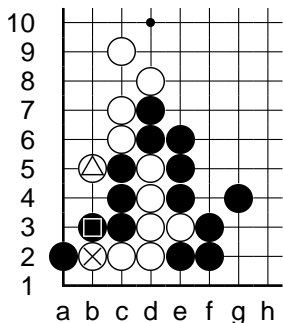following position such a hypothetical field is used to explain the optimal move which is a sacrifice.
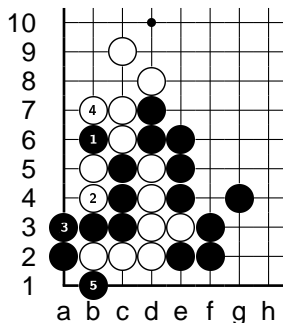

Diagram 5.
⬛ to move.

Diagram 6.
⚫ lives.

In diagram 5 the aim of Black is to make its corner chain ⬛ alive which can only happen by capturing one of the white chains ⊗ , △ . But neither one can be captured directly by playing only in its vicinity (⚫ on b6 would be captured by ◯ on b7 and ⚫ on the first row would be answered by ◯ on b4 and be too slow). Nevertheless, a static evaluation function modeling an influence field, for example the one described in section IV, would give small but nonzero values for Black's strength around a6, b6 and on the first row at b1, c1, d1. If modeled correctly these influence values alone should be too low to indicate the death of the white chains ⊗ , △ individually but if all influence is added up and increased by a value equivalent to the right of moving next then the total should be enough to indicate life for ⬛ .

The remaining question is how to convert this prospect of life for Black into a good first move. Also here the influence field helps. If this field is to be meaningful then it should not change erratically from one move to the next, except at the very end when the position becomes completely settled and the value jumps to one of the two extremes. So Black can not expect to change the total sum of all Black strengths through a clever move. But what Black can expect to do is to shift the distribution of its influence. To succeed in this example, Black needs to bundle all its influence onto its weakest white neighbour chain which is ⊗ , i.e. to move its small influence at a6, b6 towards the chain ⊗ to weaken it further. The result of the sequence in diagram 6 is that White now has total control of the points a6, b6 and Black in exchange gets one extra move towards catching ⊗ which is enough in this case.

This principle of bundling influence to overcome a threshold strength in a local target area in order to live, link, kill or cut, explains all sacrifice moves, not only in this example and not only in life and death problems. This is a good guidance not only for players but also for computer programs to formulate tactical and strategic aims, being bound on one hand by a stable total sum of influence but on the other hand being allowed to shift influence and to focus it to overcome threshold values locally for living and killing. These threshold values result from the discrete requirement of two eyes for life and the discrete nature of the capture rule capturing all stones of a chain at once.

A variation of this principle is to look for moves which make own weak stones good sacrifice stones, aiming to move the remaining strength of these stones to a distant and more important area.

In some games the principle of collecting thinly spread potential through playing 'light', i.e. through playing stones which have a good chance of being sacrificed later, is not only a tactical concept but a strategic one. When a professional player gives a strong amateur player many handicap stones then Black can not simply be fooled, the only way for White to win is to collect all potential on the board by playing light and probably sacrificing stones.

But also in even games thinly spread influence/potential is an important concept. Influence may be shifted around to force the opponent to become very strong on one side of a local area in order to gather own strength and be better prepared to attack on the opposite side of that area, in other words to create imbalance in the opponents position. This is known as a proverb: Attach against the stronger stone ([11], p. 121).

REFERENCES

[1] B. Brügmann, "Monte Carlo Go," 1993, preprint
    ftp://ftp.cgl.ucsf.edu/pub/pett/go/ladder/mcgo.ps.
[2] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2/3, pp. 235–256, 2002.

[3] L. Kocsis and C. Szepesvári, "Bandit Based Monte-Carlo Planning," in *Machine Learning: ECML 2006, Lecture Notes in Artificial Intelligence 4212*, M. S. J. Fuernkranz, T. Scheffer, Ed. Springer, 2006, p. 282293.

[4] B. Bouzy and B. Helmstetter, "Monte-Carlo Go developments," in *Advances in Computer Games. Many Games, Many Challenges. Proceedings of the ICGA / IFIP SG16 10th Advances in Computer Games Conference*, J. van den Herik, H. Iida, and E. Heinz, Eds. Kluwer Academic Publishers, 2004, pp. 159 – 174.

[5] S. Gelly, "MoGo home page,"
`http://www.lri.fr/˜gelly/MoGo.htm`.

[6] O. Teytaud, "MoGo: a software for the Game of Go,"
`http://www.lri.fr/˜teytaud/mogo.html`.

[7] R. Coulom, "Crazystone home page,"
`http://remi.coulom.free.fr/CrazyStone/`.

[8] C. Lee, M. Wang, G. Chaslot, J. Hoock, A. Rimmel, O. Teytaud, S. Tsai, S. Hsu, and T. Hong, "The computational intelligence of MoGo revealed in Taiwan's computer go tournaments," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 1, no. 1, pp. 73–89, 2009.

[9] T. Wolf, "Positions in the game of go as complex systems," in *Proceedings of 2009 IEEE Toronto International Conference (TIC-STH 2009) - Science and Technology for Humanity*. Toronto, Canada: IEEE, 2009, pp. 222–229.

[10] T. Hall and J. Fairbairn, "GoGoD Database," 2007, CD-ROM with $50,000$ professional games, `http://www.gogod.co.uk`.

[11] O. Hideo, *Opening Theory Made Easy*. The Ishi Press, 1992.