

Generalized Widening

Tristan Cazenave

presented by Martin Müller

Contents

- Threat based search methods
- Generalized threats
- Iterative Widening, Generalized Widening
- Application to open-ended Tsume-Go
- Comments

Background

- Tristan's work: use threats to improve heuristic search
- Previous papers: Generalized threats search (GTS), Iterative Widening (IW)
- Lambda-search - Thomas Thomsen

Using Threats

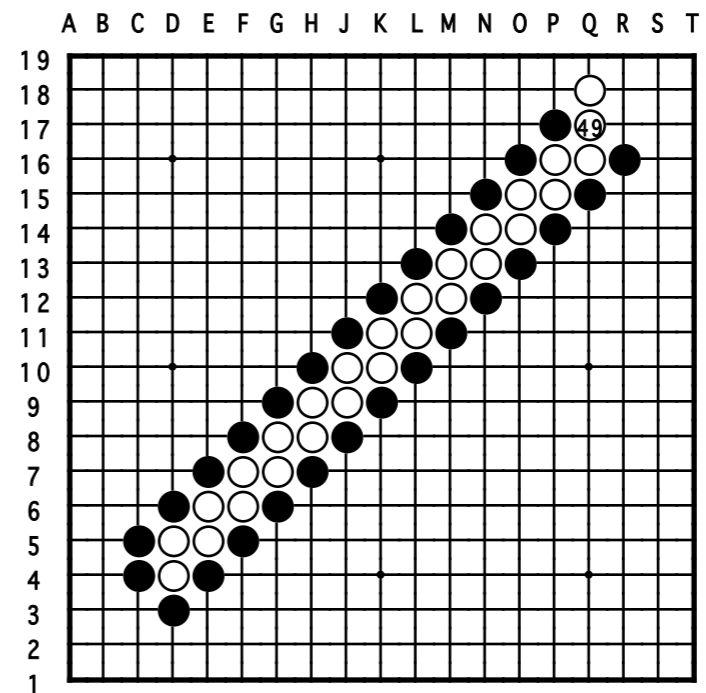
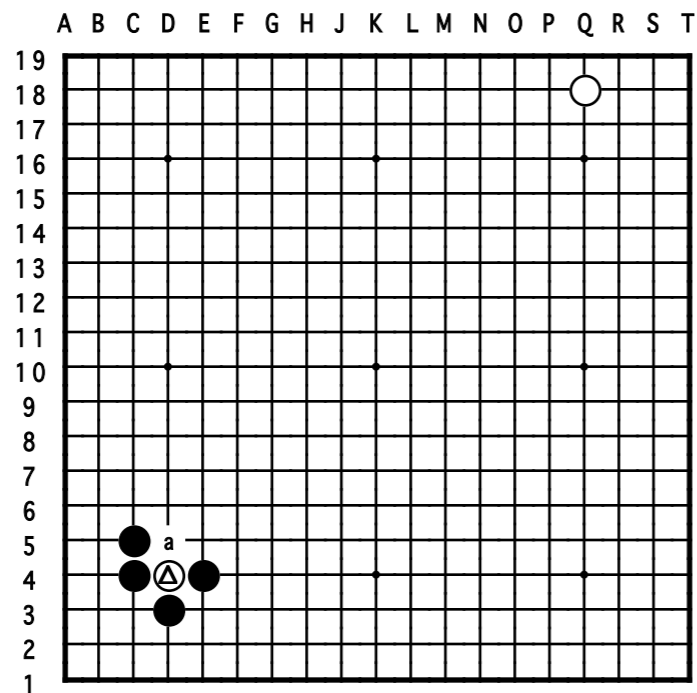
- Example: forced moves
- Example: ladder
- Example: forcing an extra approach move

Forced Move

- Example: in Kishi's solver
 - Opponent threatens to make eye
-> play there

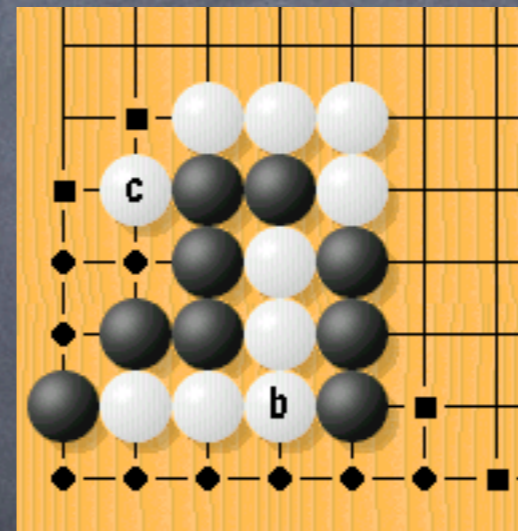
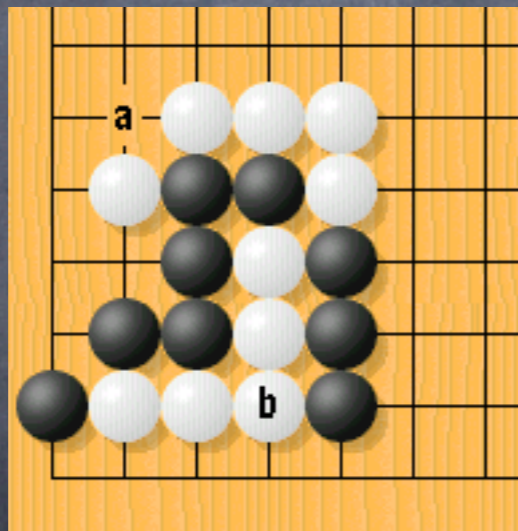
Ladder

- Trace path of ladder
- Know set of moves that can 'break' ladder



Extra Approach Move

- Example by Kierulf/Thomsen



Common Ideas

- Heuristic for Attacker's move selection
- Restrict to playing only certain types of threats (e.g. limit on depth, number of liberties)
- Advantage: strong pruning
- Verify that threat exists at opp. move

Iterative Widening

- Simple idea: iteratively increase number of moves considered
- E.g. allow more types of threats
 - Start with simple threats
 - If attacker fails, re-search with more threats

Example: order of threats

- Order of threat: number of consecutive moves attacker must play to win

In order for a generalized tree to be a generalized threat, it has to fulfill a special property: at each node of the tree that has a left and a right branch, the left subtree has to be included in the left subtree following the right branch.

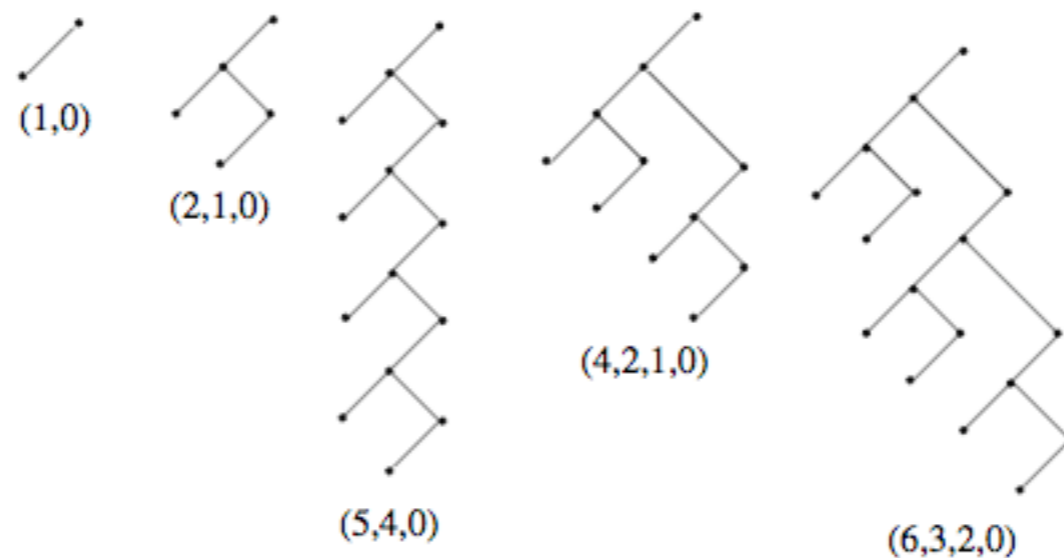


Fig. 2. Some trees representing generalized threats.

Multiple Iterative Widening

- max node:
 - start with smallest threat
 - increase until max. threat reached, or time exceeded, or node lost
- Example: (1,0) threat: develop only min nodes where a single max move threatens a direct win

Multiple Iterative Deepening

- Standard technique
 - iterative deepening search at all max nodes in search tree

Generalized Widening

- For smaller threats: complete it. deepening search until time-out or result known
- For the final threat assoc. to node: stop at search depth assoc. to node

Result values

- Win
- WinByKo
- NoThreat: threat must be increased to possibly find win
- LostByKo
- Lost

Code

```
MaxNode (alpha, beta, threat, depth) {
    eval = evaluation ();
    if (isTerminal (eval) ||
        (depth == 0) ||
        !moreTime ())
        return eval

    if (transpo (depth, alpha, beta,
                res, threat))
        return res

    Generate max moves

    // Multiple Iterative Widening
    for (t = 0;
        ((t <= threat) &&
         (res < MaxEval ()) &&
         (res > MinEval ()) &&
         moreTime ());
        t++) {
        res = eval

        // Multiple Iterative Deepening
        currentDepth = MaxDepth
```

```
        if (t == threat)
            currentDepth = depth
        for (d = 1;
            ((d <= currentDepth) &&
             (res > NoThreat) &&
             (res < MaxEval ()) &&
             moreTime ());
            d++) {
            res = MinEval ()
            alphaTemp = alpha
            betaTemp = beta
            for all max moves
                if (alphaTemp < betaTemp) {
                    play move
                    r = MinNode (alphaTemp,
                                betaTemp, t, d-1)
                    if (r > res) {
                        res = r
                        if (res > alphaTemp)
                            alphaTemp = res
                    }
                    undo move
                }
            }
        }
    }
    return res
}
```


Another view point

- See as quiescence search
 - Lower-level threats are quiescence search for higher-level threat
 - Prefers long sequence of "simple" moves over short sequence of "complicated" moves (ladder vs geta)

Application to Life+Death

- Group: include blocks that share liberties, also opp. blocks
- "Skin": add liberties, second-order liberties, weak opp. blocks, surrounded opp. blocks
- Open boundary problems
- Search "nearby" moves

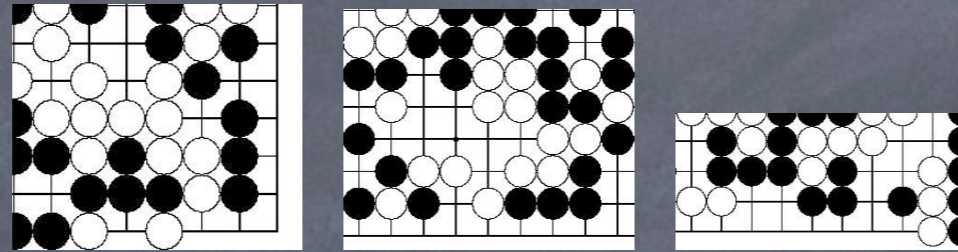
Move selection

- Specialized generator for order 1 threats (direct wins)
- General: liberties, opp. liberties with restrictions, second-order liberties

Experiments

- gw(t) Generalized widening, max. threat t
- abgw(t) Try gw(t), if fails do alphabeta
- iw Iterative Widening
- gts Generalized threats search

Results



<i>Algorithm</i>	< 0.1s		< 1s		< 10s	
	<i>solved</i>	<i>time</i>	<i>solved</i>	<i>time</i>	<i>solved</i>	<i>time</i>
$\alpha\beta$	13	3.69	21	30.71	25	250.50
$\alpha\beta gw(4)$	19	3.46	24	26.89	26	235.91
$\alpha\beta gw(3)$	19	3.43	24	26.89	27	235.50
$\alpha\beta gw(2)$	19	3.46	23	27.26	26	238.46
$\alpha\beta gw(1)$	20	3.38	21	28.75	24	262.02
$\alpha\beta gw(0)$	14	3.72	19	32.57	25	252.92
$gw(3)$	18	3.51	24	26.30	26	198.46
$gw(2)$	17	3.35	22	20.08	24	131.61
$gw(1)$	18	2.78	21	10.94	21	40.79
$gw(0)$	16	1.31	18	3.66	18	14.39
$iw(3)$	16	3.56	21	29.04	23	228.10
$iw(2)$	16	3.30	22	20.26	22	145.93
$iw(1)$	16	2.66	21	11.25	21	36.93
$iw(0)$	16	1.04	18	3.32	18	13.17
$gts(3)$	10	4.06	12	36.37	19	288.66
$gts(2)$	11	3.64	18	25.31	21	176.57
$gts(1)$	11	2.96	20	13.40	21	64.26

Comments

- Relation to df-pn?
- Strengths and weaknesses?