On Relating Local and Global Factors: A Case Study from the Game of Go

Jay Burmeister α β, Janet Wiles α β, Helen Purchase α
α Department of Computer Science β Department of Psychology
The University of Queensland, QLD 4072, Australia
jay@cs.uq.edu.au
http://www.psy.uq.edu.au/~jay/research/research_page.html

Abstract

Traditional artificial intelligence (*AI*) approaches to programming the game of Go are based on the translation of local information (modelled by pattern recognition processes) to global symbolic form (modelled by rule-based systems) for access by symbolic reasoning processes. In this paper we explore the converse process - seeking to relate local and global factors by integrating global factors into a local representation which can then be accessed by symbolic reasoning processes.

We demonstrate one method for such an integration - the interaction of bottom-up and top-down processing. The algorithm we use in our simulation integrates global factors by directly modifying the numeric information contained in the local representation of the board.

We use Go as an example of a domain with the characteristic that local and global factors cannot be identified independently of each other. Thus, to form a representation of a Go board requires an interaction between bottom-up processing (to identify local factors) and top-down processing (to identify global factors). In the final section we briefly relate these constraints to other domains.

1.0 Introduction

Influence functions are simple heuristics that are often used to evaluate board positions in Go[1] playing programs. Influence functions propagate patterns of positive and negative numeric values from each stone on the board to show how much control they exert on surrounding points. Typically, white stones generate negative influence and black stones generate positive influence. The influence values are summed at each point to give an overall value which indicates the relative size of black or white control (influence) for each point. *Influence maps* (see Figure 1) contain the numeric influence values for a given board configuration.

Influence functions are usually distanced-based with the influence exerted by a stone decaying as distance increases. There are various ways of propagating influence: onto unoccupied points [4] or onto both occupied and unoccupied points [7], [8].

Influence functions are an example of bottom-up processing which has a local scope and therefore do not encode global factors. This reduces their effectiveness as heuristic devices since they are used by symbolic processes to make decisions at the global level. There are two ways in which influence functions can be modified to overcome this deficiency.

The first is to incorporate the global factors that affect their results into the algorithm which computes the influence values. This approach will not work since knowing which global factors affect the influence function can only be determined when the influence map is used (see Chalmers, French & Hofstadter [3] for a detailed treatment of the problems associated with forming representations of global information before it is used).

1. For a brief introduction to Go rules and programs, see Burmeister & Wiles [1].

The second is to integrate global factors into the local representation that results from the influence function. In this paper we report on a simulation which demonstrates that global factors can be integrated into a local representation using a task from the game of Go.

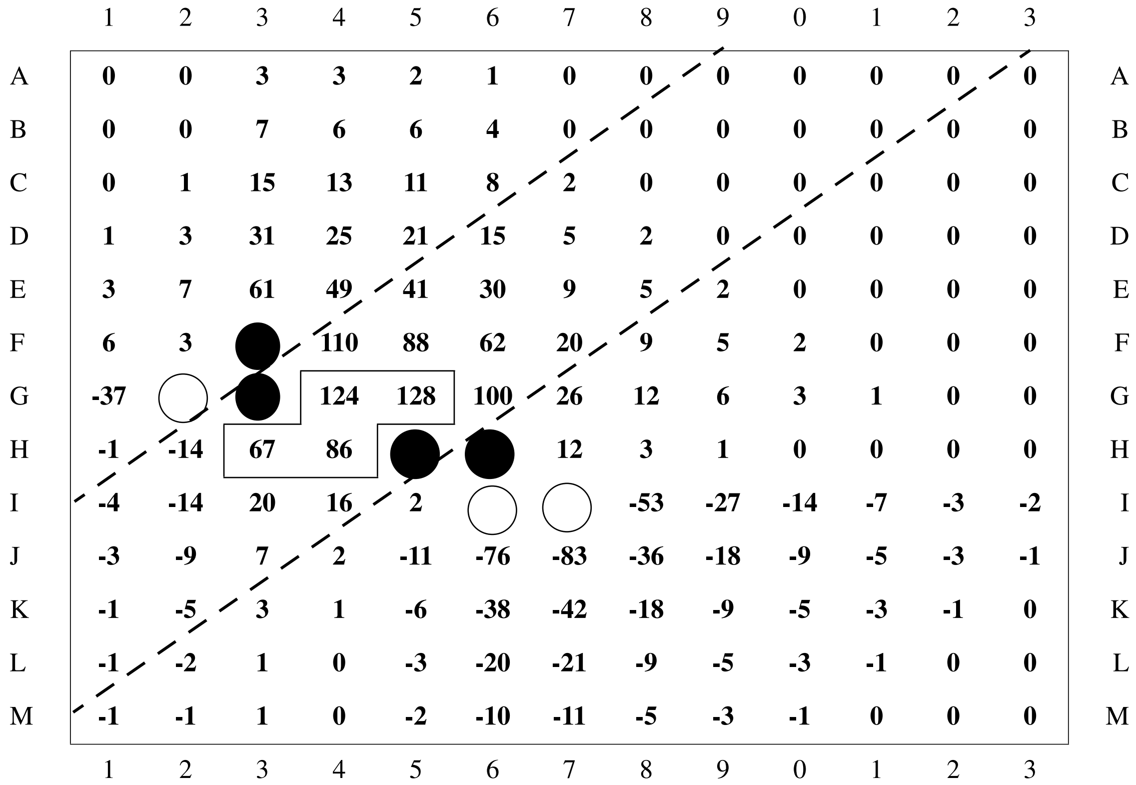

Figure 1 Influence map for the initial board position. A kogeima link exists between the black stones at (G,3) and (H, 4). The ladder-points are contained between the two dashed lines. Connection points are shown enclosed i.e. (H,3), (H,4), G,4) and (G,5).

2.0 Connectivity and Ladders

Connectivity is a (computational) measure of how likely it is that two stones can be successfully linked together to form a *string* (set of horizontally and vertically adjacent stones). One of the heuristic ways of determining the level of connectivity between two strings is the use of influence values. The stronger the influence values on potential paths between two stones, the more likely it is that they can be linked into a single string whereas the weaker the influence values between stones the more likely that the opponent will successfully impede their connection.

A *threshold of connectivity* can be set by a programmer (depending on implementation specific details) which will serve to classify links between strings as either secure or threatened. If there is a path of unoccupied board points between two stones (called *connection points*) whose influence values do not fall below the threshold of connectivity, the link can be considered to be secure. The link is considered to be threatened if every potential connection path between two stones has at least one point whose influence value is below the threshold of connectivity.

One particular sub-task of Go that requires the integration of local and global factors is the mental prediction of stone placement comprising *ladders* to determine their outcome (readout). Ladders result from a capturing race and are a repeating structure. Depending on context, ladders can result from an attempt to stop the connection of two stones which are connected by a *kogeima* link (Figure 1). Influence functions, although useful for determining local connectivity, can not be used in isolation to evaluate the outcome of a ladder. The importance of a *ladder-breaker* stone (see section 3) outside the local scope of an influence function can only be determined by top-down processing and is difficult to evaluate other than at a global level.

The simulation demonstrates how global factors relating to ladder-breaker stones at varying distances can be integrated into the local representations that result from an influence function.

3.0 Simulation: Ladder-breakers and Influence Functions

The influence values for a board configuration containing the beginning of a ladder was calculated to provide a baseline for comparison (Figure 1) as were the influence values for additional board positions containing ladder-breaker stones in three positions (close, medium and far) at increasing distances from the stones in the initial configuration.

Selected modification points were modified in the additional board positions according to a context-sensitive algorithm. The modifications are shown to overcome the local scope of the influence function and integrate global factors relating to the presence of ladder-breaker stones into the local representation (i.e., the influence map).

3.1 Influence Function

An influence function based on the one used by Chen in [4] was developed as a result of previous simulations which are described in [2].

The influence function was implemented in AKCL Lisp and the influence exerted from stone *st* to point *pt* was:

$$influence(st, pt) = m * f^{(d(st,\, pt)\, -\, 1)}$$

where:

m, the maximum value = 64,

f, the decay factor = 0.5,

d is the distance between *st* and *pt* in unoccupied points, and

influence < 1 was ignored (i.e. d(*st*, *pt*) > 7 was ignored).

The influence propagated from each white and black stone was summed at each point to give the final influence value for every unoccupied point on the board. Black stones propagated positive influence whilst white stones propagated negative influence.

3.2 Initial Position

A 13 x 13 board position is shown in Figure 1. The link between the black stones at (G,3) and (H, 5) is called a kogeima link. The Black player must play two stones on the connection points between (G, 3) and (H, 5) to physically link the two stones that are kogeima linked, and has three options to achieve this: (H, 3) and (H, 4); (G, 4) and (H, 4); (G, 4) and (G, 5).

3.3 The Ladder-breaker Stones

Ladder-breaker stones were added to the initial board position in one of three positions - *close* (F,6), *medium* (D, 8) and *far* (B,10).

3.4 Modification Points

The four connection points and the inner two lines of *ladder points* (see Figure 1) up to the ladder-breaker stone were chosen to be *modification points*. The selection of these points reflects domain specific expert knowledge and is discussed in more detail in [2].

3.7 Threshold of Connectivity

The threshold of connectivity is a domain specific value that would be empirically determined in a Go program. It was set to +30 for Black connectivity in the simulation.

3.6 Modifying Influence Values

The influence values at the modification points were modified in a context-sensitive manner. The influence value at each modification point was multiplied by a factor depending on whether it was positive or negative.The formula used was:

if *inf* * *colour* > 0 then
inf' = 1.2 * *inf*
else
inf' = 0.2 * *inf*
where:
inf is the initial influence value of a particular point,
inf' is the modified influence value of a particular point (rounded up), and
colour is +1 if ladder-breaker is black and -1 if white.

3.7 Results and Discussion

Influence maps were generated for each ladder-breaker position. From the maps, it was observed that as the position of the ladder-breaker stone was moved further away from the kogeima link (i.e., from close to medium to far), it had a decreasing effect on the influence values at the four connection points. The extent of the affect of influence propagated from the ladder-breaker in the far position is shown in Figure 2.

	1	2	3	4	5	6	7	8	9	0	1	2	3	
A	**0**	**0**	**3**	**2**	**0**	**-3**	**-8**	**-16**	**-32**	**-64**	**-32**	**-16**	**-8**	A
B	**0**	**0**	**6**	**4**	**2**	**-4**	**-16**	**-32**	**-77**	○	**-64**	**-32**	**-16**	B
C	**0**	**1**	**15**	**12**	**9**	**4**	**-6**	**-19**	**-38**	**-64**	**-32**	**-16**	**-8**	C
D	**1**	**3**	**31**	**25**	**20**	**13**	**0**	**-7**	**-16**	**-32**	**-16**	**-8**	**-4**	D
E	**3**	**7**	**61**	**49**	**41**	**6**	**1**	**1**	**-6**	**-16**	**-8**	**-4**	**-2**	E
F	**6**	**3**	●	**110**	**18**	**12**	**19**	**7**	**1**	**-6**	**-4**	**-2**	**-1**	F
G	**-37**	○	●	**25**	**26**	**100**	**26**	**11**	**4**	**-1**	**-1**	**-1**	**0**	G
H	**-1**	**-14**	**13**	**17**	●	●	**12**	**3**	**0**	**-2**	**-1**	**0**	**0**	H
I	**-4**	**-14**	**20**	**16**	**2**	○	○	**-53**	**-27**	**-15**	**-7**	**-3**	**-2**	I
J	**-3**	**-9**	**7**	**2**	**-11**	**-76**	**-83**	**-36**	**-18**	**-9**	**-5**	**-3**	**-1**	J
K	**-1**	**-5**	**3**	**1**	**-6**	**-38**	**-42**	**-18**	**-9**	**-5**	**-3**	**-1**	**0**	K
L	**-1**	**-2**	**1**	**0**	**-3**	**-20**	**-21**	**-9**	**-5**	**-3**	**-1**	**0**	**0**	L
M	**-1**	**-1**	**1**	**0**	**-2**	**-10**	**-11**	**-5**	**-3**	**-1**	**0**	**0**	**0**	M
	1	2	3	4	5	6	7	8	9	0	1	2	3	

Figure 2 Influence map after modification with the ladder-breaker stone in the far position (B,10). Points modified in the simulation are shown enclosed in a solid line. The area affected by the influence of the ladder-breaker stone is to the right and above the dashed line. Notice that the four connection points are outside this area i.e. their influence values are unaffected by the influence propagated by the ladder-breaker.

The influence values of the connection points in the initial position (Figure 1) are not affected by the influence of a ladder-breaker in the far position and therefore any decision regarding whether or not to try to connect the kogeima linked stones based on the corresponding influence map is likely to be misguided. Thus, it can be seen that local influence functions do not encode global factors associated with distant ladder-breaker stones.

The values of connection points for the close, medium and far positions have been summarized in Table 1. The entire influence map with the ladder-breaker stone in the far position is shown in Figure 2.

Initial	**Close (F,6)**	**Medium (D,8)**	**Far (F,10)**
124 128	**22 19**	**25 25**	**25 26**
67 86	**13 16**	**13 17**	**13 17**

Table 1 Influence values for the connection points for the initial position and for the close, medium and far positions of the ladder-breaker stones. (See Figure 1 to locate the connection points on the board).

4.0 General Discussion and Conclusions

In this study we have provided an existence proof that it is possible to integrate global factors into a local representation and demonstrated one method to achieve such an integration. In summary, we have shown how local and global factors can be related in the formation of a representation by the interaction of bottom-up and top-down processing.

As cognitive scientists we are interested in understanding the issues involved in relating local and global factors. Our initial interest in this area was due to the Copycat project [3] which makes analogical mappings between letter strings using an interaction between bottom-up and top-down processing. We chose Go as a domain because the determination of local and global factors cannot be done independently of each other. The computational lessons learned in Go can be applied to other more complicated domains which share this characteristic.

Natural language processing (*NLP*) is a domain in which local and global factors are difficult to determine independently. One example illustrating that local and global factors cannot be independently determined in NLP examined by Rumelhart [6]. Rumelhart described an interactive model of reading which employed an interaction between bottom-up and top-down processing. In his model, the perception of an ambiguous hand-written word was shown to depend on both the syntactic and semantic environment in which it occurs. Possible candidate words could be generated by bottom-up processing (i.e., visual similarity), however, regardless of the relative likelihood such processing provided for each candidate, a final selection could not be made until top-down processing determined which candidate also best fulfilled the syntactic and semantic constraints.

Visual processing is another domain in which the independent determination of local and global factors is difficult. Purely bottom-up and top-down models of object recognition have been criticised on the following grounds: bottom-up models identify objects by inference from their constituent features; top-down models are hypothesis driven and try to “fit” the data to the hypothesis. Pomerantz [5] showed that the familiar Gestalt maxim, “the whole is more than the sum of the parts”, bears out since subjects perceived a different object after the orientation of the constituent features of an object are rearranged. Such a phenomenon is difficult to explain in a bottom-up processing model. Rearrangement does not pose a difficulty for top-down models, however, the origin of the hypothesis does. Pomerantz proposed context to be the most important source for hypotheses and a secondary source to be bottom-up analysis of the input signal [5]. Some competing hypotheses can be ignored if they do not "fit" a quick analysis of the data leaving the remaining hypotheses to be analysed by further top-down processing. We would argue that context is simply another way to refer to global factors and would therefore have to be determined by top-down processing anyway. Thus object recognition can be best modelled by the interaction of bottom-up and top-down processing.

The important consideration illustrated by this study is that local representations resulting from bottom-up processing should not be formed at a level which requires global factors until those global factors are available. Once top-down processing reveals relevant global factors, they should be integrated into the local representation. The new local representation may then facilitate a resumption in bottom-up and/or top-down processing. Bottom-up processing may resume until once again top-down processing is required to make global factors available. Alternatively, top-down processing may use the new local representation to determine new global factors. The bottom-up and top-down processing may be performed either in serial or in parallel. Thus, it is possible that the process of integrating global factors into a local representation could be a recursive process ultimately resulting in a final representation being formed.

Acknowledgements

We thank the Department of Computer Science at the University of Queensland for the facilities to carry out this project and for providing a Summer Research Scholarship to the first author and the University of Queensland for subsequently providing a University of Queensland Postgraduate Research Scholarship to the first author.

References

[1] Burmeister, J. and Wiles, J. The Challenge of Go as a Domain for AI Research: a Comparison Between Go and Chess. In the *Proceedings of the Third Australian and New Zealand Conference on Intelligent Information Systems*, 1995.
(Available on the Internet at http://www.psy.uq.edu.au/~jay/papers/go-vs-chess.ps.Z).

[2] Burmeister, J., Wiles, J. and Purchase, H. The Integration of Cognitive Knowledge into Perceptual Representations in Computer Go. To appear in the *Proceedings of the Second Game Programming Workshop in Japan*, September 1995.
(Available on the Internet at http://www.psy.uq.edu.au/~jay/papers/c-p.integration.ps.Z).

[3] Chalmers, D. J., French, R. M., and Hofstadter, D. R. High-level perception, representation, and analogy: a critique of artificial intelligence methodology. Technical Report 49, Center for Research on Concepts and Cognition, Indiana University, Bloomington, Indiana, March 1990.

[4] Chen, K. Group identification in computer Go. In D. N. L. Levy and D. Beal, (Eds.), *Heuristic Programming in Artificial Intelligence: the First Computer Olympiad*, 195 - 210. Ellis Horwood, Chichester, 1989.

[5] Pomerantz, J. Perceptual organization in information processing. In M. Kubovy and J. Pomerantz, (Eds.), *Perceptual Organization*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1981.

[6] Rumelhart, D.E. Toward an interactive model of reading. In S. Dornic (Ed.), *Attention & Performance VI*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1977.

[7] Ryder, J. L. *Heuristic Analysis of Large Trees as Generated in the Game of Go*. PhD thesis, Department of Computer Science, Standford University, August 1971.

[8] Zobrist, A. L. A model of visual organisation for the game Go. In *Proceedings of the Spring Joint Computer Conference*, **34**, 103 - 112, 1969.