Learning Impasses in Problem Solving

J.P.E. Hodgson
Center For Machine Learning
Department of Mathematics and Computer Science
St. Joseph's University
Philadelphia. PA 19131
jhodgson@sju.edu

Abstract

Problem Solving systems customarily use backtracking to deal with obstacles that they encounter in the course of trying to solve a problem. This paper outlines an approach in which the possible obstacles are investigated prior to the search for a solution. This provides a solution strategy that avoids backtracking.

Introduction

Many weak methods of problem solving are based upon the idea that a problem can be solved by choosing a sequence of goals and satisfying them in some order. GPS (Newell and Simon 1972) was amongst the first to set out this approach. Since then the work of Ernst and Goldstein (Ernst and Goldstein 1982), Korf (Korf 1985), and Guvernir (Guvernir 1987) has built upon this idea. The culmination of this kind of approach is, in some ways, the Soar system, which through the creation of a large production system with learning capabilities is able to incorporate many of the weak problem solving systems into a single system.

If one compares Soar and Korf's system they take quite distinct approaches to the problem of what should be learned and when it should be learned. Korf's system is able to specify in advance exactly what macros it needs to learn. This will yield benefits in the system's ability to determine which macro to use at a given point in the solution, at the price of requiring long searches for some of the more complex macros. Soar on the other hand learns only the solutions to the difficulties that actually arise. This conservative attitude toward learning means that the system can encounter problems in matching expensive chunks that do not arise in Korf's situation.

This paper looks for a half way house between these two strategies. We would like to obtain the benefits of easier pattern matching afforded by Korf's system without having to pay the price of the large amount of search that his system needs. Our approach is to show that for a substantial number of problems one can anticipate the impasses that will be encountered by a problem solver. These can then be modeled and solved in small pieces of the larger problem, thus avoiding the deep searches required in Korf.

Problems, Strategies and Impasses

We review briefly the definitions that we will need. A consensus as to the appropriate definitions seems to be emerging (Banerji 1983), (Benjamin et al. 1989), and (Niisuma and Kitahashi 1985). Our definitions follow this trend. Some of them have appeared previously in (Hodgson 1989).

Problems and Subproblems

Our definition of a problem is based upon the idea of an action.

Definition 1 *A* free problem P *is a triple* (S, Ω, a) *where* a *is a partial map*

$$a : S \times \Omega \rightarrow S$$

The set S is called the state space of the problem and the set Ω is the move set of the problem. The map a represents the effect of the moves on the state space. The effect of a move ω on the state s is to give the state $a(s, \omega)$. The element $a(s, \omega)$ fails to exist precisely when (s, ω) is not in the domain of a; that is when ω cannot be applied to the state s. A sequence $\Sigma = (\omega_1, \dots, \omega_k)$ of moves on P is called *admissible* at s if the composition

$$a(s, \Sigma) = a(a(\dots(s, \omega_1), \omega_2) \dots, \omega_k) \dots)$$

exists.

We need a notion of maps between problems.

Definition 2 *Given two problems* $P_1 = (S_1, \Omega_1, a_1)$ *and* $P_2 = (S_2, \Omega_2, a_2)$*, with a pair of maps* $f : S_1 \rightarrow S_2$ *and* $g : \Omega_1 \rightarrow \Omega_2$ *The pair (f,g) defines* a strict **homomorphism** $F : P_1 \rightarrow P_2$ *provided that*

1. *Given two points* s_1 *and* s_2 *such that* $f(s_1) = f(s_2)$*, then if the move* ω *applies to* s_1 *it also appplies to* s_2 *and*
2. *The equation*

$$f(a_1(s, u)) = a_2(f(s), g(u))$$

is satisfiea in the sense that whenever the right-hand side exists so does the left-hand side.

A strict homomorphism F is a **monomorphism** *if the underlying maps f and g are one to one.*

We now turn to the notion of (strict) isomorphism.

Definition 3 *Two problems P_1 and P_2 are* **strictly isomorphic** *if there exists a pair of mutually inverse strict homomorphisms $F : P_1 \Rightarrow P_2$ and $H : P_2 \Rightarrow P_1$ between them.*

We can use monomorphisms in a natural way to define subproblems.

Definition 4 *Let $P = (S, \Omega, a)$ be a problem. A problem $P_0 = (S_0, \Omega_0, a_0)$ is a* **(strict) subproblem** *of P if there exists a problem monomorphism $(f, g) = F : P_0 \Rightarrow P$.*

It is worth noting that the requirement that the underlying state map be a monomorphism has the effect that even the weaker definitions of homomorphism such as the weak homomorphism of Niisuma and Kitahashi (Niizuma and Kitahashi 1985) lead to the same subproblems.

As an example of the concepts developed here we can take the sliding tile pussles. In particular we might take the fifteen pussle. The state space is then the set of all legal arrangements of the fifteen tiles and the blank in the 4 × 4 array. The move set can be given by the set $\Omega = \{U, D, L, R\}$ where the letter indicates the direction in which a tile is moved. A typical subproblem can be obtained by restricting one's attention to the tiles in the top half, (assuming that the blank lies in the top half). Moves on this subproblem are restricted to be those in which the blank remains in the top half of the array.

Strategies

So far we have not recognised that problems are supposed to represent things that are to be solved. To do this we define a **problem instance** for a problem P as a triple (P, σ, G) where σ is a state of P called the start state, and G is a subset of the state space called the goal set. A *solution* to the problem instance is a sequence Σ of moves which is admissible at σ and such that $a(\sigma, \Sigma) \in G$.

Informally a strategy is a sequence of intermediate subproblem instances between the initial state and the goal. We can distinguish two classes of strategies. In the first the successive state spaces overlap; we call these ample strategies. In the other the successive state spaces are disjoint; we call these abutment strategies.

Definition 5 *An* **ample strategy** *for a problem instance $(S, \Omega, a, \sigma, G)$ is a sequence $\{P_0, \ldots, P_k\}$ of subproblems of $P = (S, \Omega, a)$ such that the state spaces of successive subproblems have non-trivial intersection, that is $S_{i-1} \cap S_i \neq \emptyset \ \forall i \in 1, \ldots, k$. Furthermore $\sigma \in S_0$ and $G \subseteq S_k$.*

An **abutment strategy** *for a problem instance is a sequence $\{P_0, \ldots, P_k\}$ of subproblems of P such that*

1. *$\sigma \in S_0$,*
2. *$G \subseteq S_k$,*
3. *$S_{i-1} \cap S_i = \emptyset \ \forall i \in 1, \ldots k$,*
4. *for each $i \in 1, \ldots k$ there exists at least one pair (x_{i-1}, x_i) of points of S such that there is a move $\omega \in \Omega$ for which $a(x_{i-1}, \omega) = x_i$.*

A solution is based upon a strategy if it is obtained by concatenating a sequence of solutions to the intermediate subproblems.

We illustrate the two kinds of strategies with examples. For our example of an ample strategy we consider the case of Fool's disk. This problem has been discussed by Ernst and Goldstein (Ernst and Goldstein 1982). It consists of four concentric rings each of which is free to rotate about the common center. Each ring has eight numbers on it, appearing at 45 degree intervals around the ring. The goal of the problem is to rotate the rings so that the sum of each radius is 12. The standard strategy is as follows:

- By using only rotations through 45 degrees, make the sum on each pair of perpendicular diameters 48. P_0 thus has the same state space as P but a smaller move set.
- By using only rotations through 90 degrees, make the sum along each diameter 24. P_1 has as state space a set of states in which the sum on each pair of perpendicular diameters is 48. The move set is again a subset of the original one.
- By using only rotations through 180 degrees make the sum along each radius 12. P_2 has as state space a set of states in which the sum along each diameter is 24 and once again the move set is a subset of the original.

This strategy, when successful (about which more later), reduces the amount of search from 8^3 moves (the center ring can be kept fixed) to 8 × 3 moves.

Our second example is an elegant solution of the five pussle that has been presented by Banerji (Banerji 1990). He observes that there is a way to represent the states of the five pussle by points on the faces of a dodecahedron. The sequence of moves that circulates the blank around the the circumference of the the pussle moves through all the states on one face. Passage from one face to another is effected by the moves that slide the blank up and down in the centre column. The strategy in this case consists of choosing the sequence of faces (each of which is a subproblem) through which one must pass from the start to the goal.

There is an important difference between these two examples. In the second example once the strategy is chosen no backtracking over the solutions to the intermediate problems is necessary but in the case of the fool's disk it may be necessary to backtrack since it is possible that the first arrangement in which the sum on all the diameters is 24 may not lead to a solution and another arrangement is needed. Niizuma and Kitahashi (Niizuma and Kitahashi 1985) give a sufficient condition for this not to occur.

Proposition 6 *Suppose that for each subproblem occurring in a strategy it is the case that any instance of the subproblem can be solved then no back tracking will be needed to construct a solution to the original problem following the strategy.*

It may seem that the restriction on the state spaces of the intermediate problems is unduly restrictive. Yet it is exactly this that is needed to avoid backtracking. Thus one aim of our approach is to find strategies for which this hypothesis is true.

Impasses

At any stage in the execution of the strategy one has a subproblem instance $(P_i, \Omega_i, a_i, \sigma_i, G_i)$ where in the case of an ample strategy G_i is $S_i \cap S_{i+1}$ or in the case of an abutment strategy G_i is the set of points of S_i from which a move to S_{i+1} is possible. We have seen that the strategy proceeds smoothly as long as these intermediate problems can be solved.

Definition 7 *An intermediate problem for which there is no solution is called an* **impasse** *for the strategy.*

This terminology follows one case of the use of the term in the Soar system, in so doing we are also following the usage of Ruby in (Ruby and Kibler 1989).

It is important to note that our definition of an impasse in a problem is dependent upon the strategy chosen to solve the problem. Thus in the Banerji solution to the five puzzle there are no impasses since each intermediate goal is attainable. By contrast in the more usual method in which the tiles are brought into position in a prearranged order there are impasses.

Learning the Impasses

Our approach to finding impasseless strategies is to improve an existing strategy by modifying the subproblems so that they do not contain any impasses. As an example we show that in the case of Fool's disk we can do this by enlarging the intermediate problems. This need not always be the case as we shall see in some of the examples that we discuss.

For the Fools' disk case we can consider the intermediate problems defined as follows:

- By using only rotations through 45 degrees, make the sum on each pair of perpendicular diameters 48. P_0 thus has the same state space as P but a smaller move set.
- Let P_1 have as state space the set of all states in which the sum on each pair of perpendicular diameters is 48. The move set is again a subset of the original one. It may contain some moves through 45 degrees.
- Let P_2 have as state space the set of all states in which the sum along each diameter is 24. The move set may contain moves through 90 or even 45 degrees.

It is clear that for these problems no backtracking into earlier problems is necessary.

Finding the Impasses

Problem solvers such as Soar (Laird et al. 1986) and the stepping stone method (Ruby 1989) find the impasses in the course of attempting to satisfy the current goal. A search procedure is then invoked to resolve the impasse and the resolution of the impasse become part of the problem solver's knowledge about the problem. This is an accurate representation of much human problem solving, but it does not tell the whole story. Often faced with a problem a human will actively consider the difficulties that may arise in the course of the resolution of the problem to see if they can be solved. One advantage of such an approach offers is that it allows one to take advantage of efficient storage techniques once one has determined that a small group of chunks will be adequate to solve the problem. This addresses in some measure the problem of expensive chunks (Tambe et al. 1990).

We give here a recognition criterion that forms the basis for an algorithm that can be used to produce impasses in problems. The criterion will be stated for the cases where the strategy is based upon the idea of reducing a set of features to their goal values. We begin by formalising this notion.

Given a problem P a feature on P is a map

$$f : S \rightarrow T(f)$$

between the state space of P and some finite set $T(f)$ called he target of of the feature. A set $\mathcal{F}$ of features is called **discriminating** if given any two state s_0 and s_1 of P there is some feature $f \in \mathcal{F}$ such that $f(s_0 \neq f(s_1)$. The set is called **adequate** for a goal G if given any state s which is not a goal state there is some feature f such that $f(s)$ is not a member of $f(G)$.

The strategy associated to an ordering $\{f_1, \dots, f_k\}$ of a set of adequate features is the sequence of subproblems $P_i = (S_i, \Omega, a_i, \sigma_i, S_{i+1})$ where S_i is the set of all states for which the features $f_1, \dots, f_{i-1}$ have goal values, a_i is the restriction of a to $a^{-1}S_i \cap (S_i \times \Omega)$. For these strategies we can give a recognition criterion for impasses.

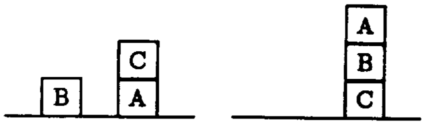
Proposition 8 *Let $P_i = (S_i, \Omega, a_i, \sigma_i, S_{i+1})$ be an intermediate problem for a strategy based upon an adequate set of features, then P_i is an impasse instance if either*

- *No move changing the value of f_i applies to σ_i, or*
- *Any sequence of moves on P that reduces f_i from σ_i must change the value of at least one of $f_1, \dots f_{i-1}$.*

From this point forward the argument goes as follows. First, find an impasse. Second, produce a "smaller" example of the same impasse. Thirdly, expand the example to a subproblem in which the impasse can be resolved. Finally, show that the problem has a strategy based upon the new set of subproblems.

Examples of Impasses

To obtain an impasse of the first kind we can turn to Sussman's anomaly in the blocks world. The impasse can be succinctly described by the following figure.



Sussman's Anomaly

Although one can get "closer" to the goal by putting B on top of C in the position on the left hand side it will be necessary to undo this since the goal of putting A on B requires that the top of A be clear. Thus no move that will achieve the desired position for A is available.

To get an example of the second kind we consider the fifteen pussle with the initial strategy of moving the tiles into position in the order $1,2,3,4,5,6,7,8,9,13,10,14,11,12,15$ (the ordering at the end is chosen to be a good one, we do not need to go this far though).

1	2	3	5
15	6		4
9	10	12	11
13	14	8	7

A Fifteen Pussle Position

In the diagram above we find an impasse when we come to try and locate tile 4. The smallest subproblem in which this impasse appears is the 2×2 upper right hand corner in the diagram where which we place 3, 5, 4, blank reading clockwise from the top right. (The choice of 5 is not significant.) This can be solved in the five pussle that is obtained when we restrict attention to the top two rows and rightmost three columns of the pussle. Furthermore we can cover the state space of the fifteen pussle with copies of the five puzzle in the way that will be detailed in the next section and obtain an impasse free strategy.

In fact the recognition criterion given in proposition 8 permits one to write a simple program that will generate impasses in both these cases. Furthermore the expansion of the subproblem described in the example of the sliding tile pussle will provide the means for resolving the impasses. This is the subject of the next section.

Atlases: Solving the problem

In this section we will describe a modified version of the notion of a strategy. In some sense it is a metastrategy in that it is designed to produce an impasse free strategy for a problem by choosing the sequence of subproblems from a set of subproblems whose image cover the whole of the state space. The basic idea is that one determines what impasses may arise in the problem and then expands them to subproblems that resolve the impasses. These impasse resolving problems are then used to cover the state space of the problem giving rise to a new strategy.

Charts

It is convenient to introduce two auxiliary notions. These are chart and atlas. The idea is that chart are pieces of a problem that are all modeled on some common subproblem. The important charts will be the ones that contain the resolutions of impasses.

Definition 9 *Let P be a problem and s a state in P. Then a* **chart** *for P based upon a problem P_0 is a problem monomorphism $P_0 \to P$ whose image contains s.*

An **atlas** for a problem P is a finite collection $\mathcal{A}$ of charts such that every point in the state space of P is in the image of some chart of $\mathcal{A}$.

We define the images of two charts $f_1 : P_1 \to P$ and $f_2 : P_2 \to P$ to be *incident* if either

1. $f_1(P_1) \cap f_2(P_2)$ contains at least one move common to both subproblems, or
2. there exists a state $s_1 \epsilon f_1(P_1)$ and a state $s_2 \epsilon f_2(P_2)$ such that there is a reversible move ω with $a(s_1, \omega) = s_2$.

The ***abstraction*** of a problem associated to an atlas is the graph whose vertices correspond to the embedded charts of the atlas with an edge between each pair of incident charts. A sequence of pairwise adjacent charts is called a *chain*.

We will want to distinguish between two types of abstraction. An abstraction in which the charts overlap will be called an *ample* atlas. One in which all the charts are incident but do not overlap will be called an ***abutment*** atlas.

We give two examples of abstractions associated to an atlas. The first is based upon the earlier solution of the five pussle. Here the charts consist of the images of the sub-problem of the five puzzle consisting of those states that are obtainable by moving the blank around the circumference of the pussle. As Banerji remarks

(Banerji 1990) this represents the accessible states of the five puzzle on the faces of a dodecahedron. The faces of this are the points of the abstraction and the edges (which correspond to the move of the blank up or down in the middle column) correspond to the edges.

We can obtain an abstraction of the blocks world by "welding together" adjacent blocks so that we have only three big blocks to consider. Each big block is itself a blocks world and the three block world already contains the generating example of the impasse.

These two examples suggest that the correct choice of an atlas will allow one to give an impasse free strategy.

The Atlas Meta-strategy

Atlases serve as abstractions of a problem. Given a problem instance and an atlas on a problem we can define a problem instance on the atlas. ~he problem is to find a chain joining a chart contai..ng the start position to a chart whose image intersec:s the goal.

Definition 10
Given an impasse $I = (P_0, \Omega_0, a, \sigma_0, G_0)$ on a problem, a chart $f : P_1 \rightarrow P$ is said to resolve *the impasse if there is a monomorphism of P_0 into P_1 and if the instance I can be solved in P_1.*

The main ideas of this paper can be summed up in the following.

Proposition 11 *Let $\{P_0, \ldots P_k\}$ be a strategy for a problem P and let $\mathcal{I}$ denote the set of impasses for this strategy. Let $\{Q_1, \ldots, Q_n\}$ be a set of charts of P such that each impasse is resolved in at least one of the Q_i. There is an atlas $\mathcal{A}$ based upon the charts Q_i whose associated meta-strategy gives impasseless strategies for P.*

The next section outlines a proof of this result and to a result on the length of the solutions that it produces.

Solutions and Their Length

The ideas required to construct the impasseless strategy are outlined below. The details have been worked out for the sliding tile puzzles, the Tower of Hanoi and the blocks world but in a manner that is somewhat problem dependent. Future work involves unifying the implementation so that it applies in a more problem independent way.

Resolving the Impasses

Let $P_i = (S_i, \Omega_i, a_i, \sigma_i, S_{i+1})$ be an impasse arising from the strategy based upon the set $\{f_1, \ldots, f_k\}$ of features on a problem P. The following sequence of steps is used to resolve the impasse.

SHRINK

The goal of this step is to remove from consideration those features that are not required to construct the impasse. In general given a set of features on a problem we can restrict to the moves that affect only these features. The required shrinking takes place by eliminating the features which are both fixed and whose value does not figure in the creation of the impasse.

ENLARGE

Moves that effect the remaining features are now added to produce a subproblem in which the impasse can be solved. At each step the move added should affect the smallest possible number of additional features.

An Example

We can illustrate this process with the example of the fifteen puzzle. We saw that an impasse can be reached when the first three tiles have been placed. The SHRINK process reduces this to an example equivalent to a three puzzle in which the tiles appear in the order 3, x, 4, blank, when read clockwise from the top left hand corner, (x denotes one of the possible tile values other than those already used.) We can then EXPAND to a five puzzle, which can be either horisontal or vertical in which the impasse is resolvable.

The next step is to determine whether there is an atlas for the problem whose charts are isomorphic to the set of subproblems obtained by resolving the impasses. If this is the case we then replace the original strategy by the following one. We suppose as before that we have a problem P with an adequate set of features $\{f_1, \ldots, f_k\}$. In addition we assume that there is an atlas $\mathcal{A}$ whose charts are isomorphic to the impasse resolving subproblems obtained by the process outlined above.

Using the same ordering of features that was used for the original strategy that produced the impasses.

1. Set as the current subgoal the reduction of the next feature to its goal value.
2. As each feature comes up for reduction find a chain of minimal length joining the current state to a state in the current subgoal.
3. Extract the move sequence joining the current point to one in which the feature has been reduced.

Since the atlas contains a resolution of all impasses this method will solve the problem whenever there is in fact a solution.

The Length of a Solution

We can now give an estimate for the length of a solution found using this method. We need some preliminary definitions.

L will stand for the maximum chain length required to perform the reduction of a feature.

D will stand for the maximum distance between two states in a chart. When a particular chart C is r· ferred to we will use $D(C)$ for the distance on th chart. Note that this number can be infinite if t: chart is an impasse chart.

N will be the number of features on the problem.

n will be the number of chains required to reduce all the features.

The first result is the following

Theorem 12 *Let P be a problem with an ample atlas and features with values of L, D, n as above. Then the algorithm given above finds a solution of length at most $L \times D \times n$.*

Proof. For each feature the length of chain required to reduce it does not exceed L, furthermore one each component of the chain the length of the move sequence required is less than D.□

The corresponding result for abutment atlases is the following. The proof is similar.

Theorem 13 *Let P be a problem with an abutment atlas and features with values of L, D, n as above. The algorithm above supplies a solution of length at most $n \times (L \times D + L - 1)$.*

Although these results are quite simple they give quite good estimates. For example in the case of the fifteen puzzle if we use the estimate of 22 as the maximum distance on the five pussle (Banerji 1990) we get an estimate of $(22 \times 3 \times 15) + 3$ for the length of a solution. A more perspicuous version of the argument yields $(19 \times 22) + 3$.

Summary and Conclusions

This paper has presented a method for solving problems that constructs the impasses associated to an initial strategy in order to be able to find a new strategy in which impasses will not arise.

The method can be applied to produce short solutions to the sliding tile puzzles as well as to the blocks world. Though the implementation is at this stage still very problem dependent. Future work will produce a version that is more general.

Acknowledgements I am grateful to Ranan Banerji for his reading of earlier drafts of this paper and for numerous helpful comments during the course of this work.

References

R.B. Banerji 1983. *Artificial Intelligence: A Theoretical Approach.* North-Holland.

R.B. Banerji 1990. Heuristics: Alternatives to minimum distance and a strange state of the five puszle. In *Intelligent Systems State of the Art and Future Directions.* (Ras and Zemankova Eds. Ellis Horwood Chichester UK .

P. Benjamin, L. Dorst, I. Mandhyan, M. Rosar 1989. An Introduction to the Decomposition of Task Representations in Autonomous Systems. Technical Report Philips Labs.

G.W. Ernst and M.M Goldstein 1982. Mechanical Discovery of Classes of Problem Solving Strategies. J ACM 29 1 1-23.

H.A. Guvenir 1987. Learning Problem Solving Strategies Using Refinement and Macro Generation. Ph.D. Thesis. Case University.

J.P.E. Hodgson 1988. Solving Problems by Subproblem Classification. Proceedings ISMIS Torino.

J.P.E. Hodgson 1989. Automatic Generation of Heuristics. In *Formal Methods in Artificial Intelligence: A Sourcebook.* Ed. R.B. Banerji. Elsevir.

R. Korf 1985. *Learning to Solve Problems by Macro-generation.* Pitman.

John Laird, Paul Rosenbloom, and Alan Newell 1986. *Universal Subgoaling and Chunking.* Kluwer Academic.

A. Newell and H.A. Simon 1972. *Human Problem Solving.* Prentice- Hall. Englewood Cliffs. NJ.

S. Niisuma and T. Kitahashi 1985. A Problem Solving Method Using Differences or Equivalence Relations Between States. *Artificial Intelligence* 25 117-151.

David Ruby and Dennis Kibler 1989. Learning Subgoal Sequences for Planning. Proc 11th ICJAI. 609-614.

M. Tambe, A. Newell and P.S. Rosenbloom 1990. The Problem of Expensive Chunks and its Solution by Restricting Expressiveness. *Machine Learning.* 5 299-348.