Multi-Dimensional Heuristic Searching

Peter C. Nelson
Dept. of Electrical Engineering
and Computer Science
University of Illinois at Chicago
Chicago, Illinois 60680

Lawrence J. Henschen
Dept. of Electrical Engineering
and Computer Science
Northwestern University
Evanston, Illinois 60208

Abstract

A heuristic improvement technique referred to as multi-dimensional heuristics is presented. Instead of only applying the heuristic between two states X_1 and X_2, when a distance estimate of X_1X_2 is needed, this technique uses a reference state R and applies the heuristic function to $\{X_1,R)$ and (X'_2,R) and compares the resulting values. If two states are close to each other, then they should also be approximately equidistant to a third reference state. It is possible to use many such reference states to improve some heuristics. The reference states are used to map the search into an N-dimensional search space. The process of choosing reference states can be automated and is in fact a learning procedure. Test results using the 15-puzzle are presented in support of the effectiveness of multi-dimensional heuristics. This method has been shown to improve both a weak 15-puzzle heuristic, the tile reversal heuristic, as well as the stronger Manhattan distance heuristic.

1 Introduction

Traditional heuristic search involves ordering state expansions relative to their estimated costs of participating in a solution. This cost is computed by a function $f = g + h$ where g is the known cost (depth) arid h is an estimate of the remaining cost or distance to the goal [Hart *et* a/., 1968]. This paper proposes a general method of improving *h.* Instead of estimating h by evaluating a given state X with respect to the goal G. several new reference states are used to gain perspective [Nelson, 1988]. The relative position of X and G among the reference states will be used to estimate the distance between X and G.

Assuming h is of a general nature and can be used to estimate the distance between any two states in the search space, then h can be used to estimate the distances from the reference states to both G and X. For each reference state R_i, a difference value $AR_i = | h(X, Ri) - h(G, R_i) |$ is computed, which is the absolute value of the difference between the estimated distances from X to R_i and from G to R_i. Note that if the goal is used as a reference state r_K, then ARk is just the traditional heuristic estimate *h.* These difference values AR_i, where $1 < i < n$ and n is the number of reference states, will be used to give a better estimate of the actual distance from the X to G. The new estimate will be referred to as H_n and will be proportional to the values of AR_1, AR_2 . .. AR_n as will be discussed in section 2. The method for combining the AR values to get H_u is based on mapping the search space into an N-dimensional space which is why H_n is referred to as a multi-dimensional heuristic or MDH. Another key part of the calculation of H_n is that the AR values are independent of each other and can be computed simultaneously on a multiprocessor architecture. Also $h(G,R_i)$ is fixed and need only be computed once for each R_i.

We would hope that a smaller AR_i value for reference state R_i would indicate that X is closer to G. The intuition is that if X is close to G, then both states should be approximately the same distance from any given reference state, thus yielding a small AR value. The farther X is from G, then the greater the AR value will be. Figure 1 illustrates this. In this example only one reference

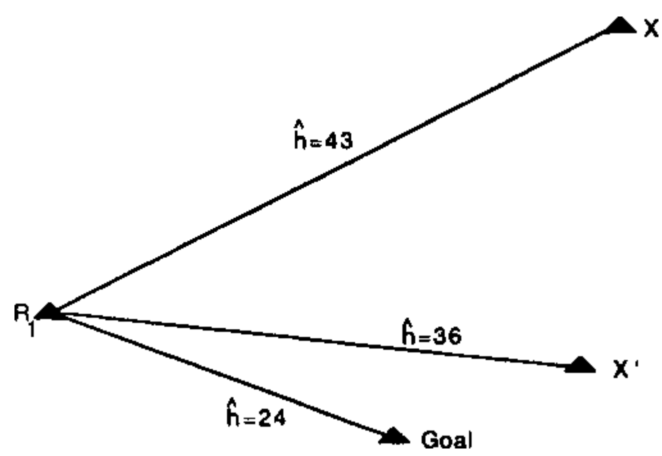

Figure 1: Ordering Nodes using Reference Nodes

node is being used. Nodes X, X', and G are estimated to be 43, 36 and 24 units away from $R_{,1}$ respectively. The ΔR_X value for X is 19 while ΔR_1, for X' is 12. With respect to R_1 we see that X' has a smaller $AR\backslash$ value than does X which is obviously farther away from G than X'. Also note that because H_1 will be proportional to ΔR_1,

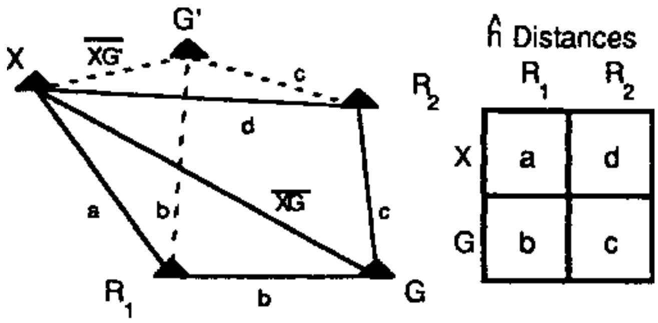

Figure 2: Planar View of Reference Nodes

$H_1(X')$ will be less than $H_1(X)$.

Of course, if X lies directly opposite R_i., from G, it may have a small ΔR_i,- and still be far away from G. For each reference node there will be such a region of declination relative to a given goal [Nelson, 1988]. The fix for this involves adding more reference nodes and positioning these reference nodes so their (estimated) distances to each other is relatively large and their (estimated) distances to the goal node varies.

The calculation of H_n and the reasons for viewing the search space as an N-dimensional space are explained in section 2. Section 3 reports on some empirical data using MDHs for the 15-puzzle. Concluding remarks and ideas for further research are contained in section 4.

2 Viewing the Search as an N-space

At this point we demonstrate why a planar view cannot be used satisfactorily to compute a distance from X to G using reference nodes and explain why the term "multi-dimensional" has been chosen to refer to this idea of using many reference states to estimate a distance to the goal. Figure 2 shows a typical case in using MDHs. The distance in question is between X and G. Estimates of the distances between the reference states to X and G are known. This planar or 2-dimensional view yields many possible values for the estimated distance between the states X and G. Two of these distance estimates are shown in figure 2. The solid lines indicate one possible layout while the dotted lines show another. Geometry can be used to explain the difficulty here even though the concept of an angle in a problem space is undefined. Because the angle values in figure 2 are not known, an infinite number of distance estimates can be found for XG by varying the angles LR_1XR_2 and LR_1GR_2 (or equivalently moving $R\backslash$ or $R_2)$ while still preserving all the distances between X and G and the reference states. If these angles are set by using h to estimate the distance from $R\backslash$ to $R_{2)}$ the resulting value of $R_1 R_2$ may lead to other inconsistencies. For example we know that R_1R_2 < a+d, but perhaps (the estimated value *of* $R_1R_2)$ $h\{R_1, R_2)$ > a $+d$. The computation of these different distances is relatively expensive and it is not known which to use as an estimate. That is, should H_2 be XG XG, or one of the other possible distance values. The

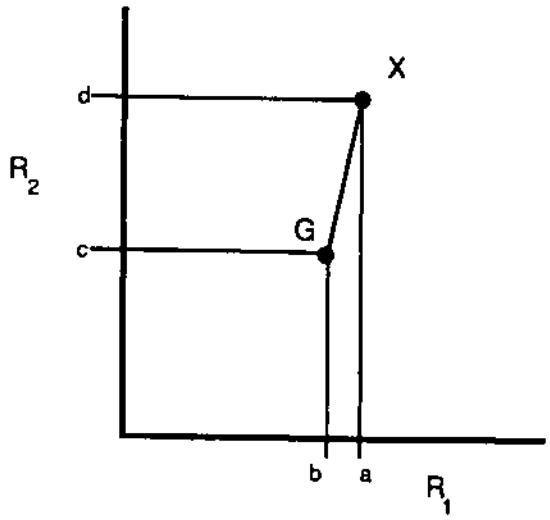

Figure 3: Using Two Dimensions

addition of a third reference state does not remedy this problem of multiple values for XG [Nelson, 1988].

This inability to uniquely determine a distance for XG in the planar view is the impetus for a different model of the search space and reference states. This new model uses the reference states to set up an N-dimerisional search space. The number of dimensions is equal to the number of reference states. The current and the goal states are mapped into this N-space and their distance estimate is computed using; the standard distance formula $\sqrt{(x_2 - x_1)^2 + \cdots + (z_2 - z_1)^2}$. The mapping of the states into the N-space is done by letting the estimated distance of the goal (current state) to any given reference point R_i be the coordinate value of the goal (current state) with respect to the R_i axis. The example shown in figure 2 would be mapped into a 2-space depicted by figure 3.

This representation solves the problems associated with the planar view. There is only one distance value associated with any two states mapped into the N-space. The computational cost of H_n is also quite reasonable. This model yields the desired property that for any given ΔR_i, if ΔR_i decreases while everything else is constant, then so does the distance estimate between the two states. This requirement was mentioned in section 1 which stated that H_n should be proportional to the AT? values.

One quest ion has been raised as to how the MDH value should be computed if the original h consists of k components. Although this would really have to be examined on a case by case basis, there are really two reasonable approaches to consider:

1. Ignore that h is a vector of components since it is still a distance estimator and can be used to estimate distances between the reference nodes.
2. Analyze each component of h to see which components, if any, can be improved by using the multidimensional scheme. For each component which can

be improved using MDHs, pick a set of (possibly distinct) reference nodes and redefine h such that the appropriate components are actually MDHs. For example let

$$\hat{h} = \hat{h}_0 + \cdots + \hat{H}_{\hat{h}_i} + \cdots + \hat{h}_k$$

where h_i is replaced with the improved MDH component H_{h_i} .

Probably most heuristics which are built with a number of different components are already quite good and may not be able to achieve much of an improvement by using the first approach which ignores the different components. It may be more likely to improve such a heuristic by focusing on improving specific components of the heuristic with MDHs. The improvement of a specific component in this case would of course be defined in terms of how this component affects the accuracy of *h.*

For any given state space problem and the *h* being used to estimate distances to the reference states, there are at least two questions that need to be answered in order to apply MDHs:

1. what should the value of n be, that is how many reference states should be used, and
2. which reference states should be used to approximate the N-space.

The next section addresses these questions.

3 Test Results

The multi-dimensional approach was tested on the 15-puzzle using the *A** algorithm as described by [Rich, 1983]. The goal of the testing was not to show that there exists an MDH which is more effective than any other existing heuristic, but rather to show that MDHs may be used to improve some existing heuristics. Therefore, initially a heuristic was chosen for the 15-puzzle which had room for much improvement, this heuristic being a tile reversal count. Later the multi-dimensional scheme was tested on a much stronger 15-puzzle heuristic, the Manhattan distance heuristic.

3.1 Tile Reversals and MDHs

The first set of tests used *h* as the number of tile reversals. A tile reversal in state a with respect to state b has the meaning that a(i) = b(j) and b(i) = a(j) where i and j are adjacent tiles [De Champeaux and Sint, 1977]. The rationale for the heuristic is that if a reversal occurs between a state and the goal, then it takes many moves to get the tile positioned correctly. The problem with tile reversals is that they do not occur often, so most of the heuristic values are 0, and the search just flattens out into a breadth first search. Usually tile reversals are one component of a more sophisticated heuristic for the 15-puzzle. Note also that this choice for *h* is general in that it may be applied to any two nodes in the 15-puzzle search space.

The reference states were picked by first generating a number of states in a random fashion from which the reference states would be chosen. Then several hundred legal states (i.e. reachable from the goal) for which the distances to the goal were known were evaluated with an MDH using all the randomly generated reference states. (These legal states were found with their distances by generating nodes in reverse from the goal node.) Ideally an MDH would have reference states that yield distance estimates proportional to the actual distances. Therefore a score was kept to determine the predictive accuracy of each reference state. This was done by first computing the average of h(x) / $H_n(x)$ for every x where x is one of the several hundred legal states, h(x) is the actual distance from x to the goal, and $H_n(x)$ is the MDH value for x using the n randomly generated reference states. Once this average is computed, h(x) / $H_n(x)$ was (retrieved) recomputed for every state x. The ratio for each state x was compared with the average ratio taken for all the states. If this ratio was close to the average ratio, then every reference node which participated in the calculation received a "good" mark. If the ratio was not close to the average ratio, then all the reference nodes which helped to calculate it received a "bad" mark. At the end the score was tallied by subtracting the number of bad marks from the number of good marks. The score for a reference state indicates whether that R_i helps or hinders the MDH in achieving the goal of yielding distance estimates proportional to actual distances. These scores were computed for each of the possible reference states and a few of these states with the lowest scores (net goodness values) were eliminated from the set of possible reference states. This process was repeated many times with each iteration eliminating possible reference states until there remained 13 possible reference states. This learning procedure, which is described more formally in [Nelson, 1988], is general and could be applied to other problem domains as long as states can be generated in reverse order from the goal state.

Figure 4 shows the results from tests run on 30 puzzles using from 1 to 13 reference states. The goal state was added as a reference state and was ordered as the first reference state, thus the 1-dimensional trial is exactly identical to a traditional *A** search using tile reversals as the *h.* The number of nodes expanded is inversely proportional to the number of dimensions used by the MDH with the exception of some relative maxima at dimensions 5 and 7. With the addition of each of the first 3 dimensions, the search space is cut in half. The graph in figure 4 shows that for these puzzles the best value for N is probably 4, since there is relatively little improvement in adding any dimension past the fourth dimension.

The initial set of tests for the tile reversal case were run on puzzles with solution paths of length 10. Although the average path lengths for the 15-puzzle is about 50, these shorter puzzles were simple enough so that every dimension was capable of finding a solution without running out of memory. Using the same 13 reference nodes more tests were run on puzzles with solution paths of length 15, or a 50% increase from the previous tests. The results were consistent with the tests run on shorter puzzles. The biggest difference is that no puzzles were solved

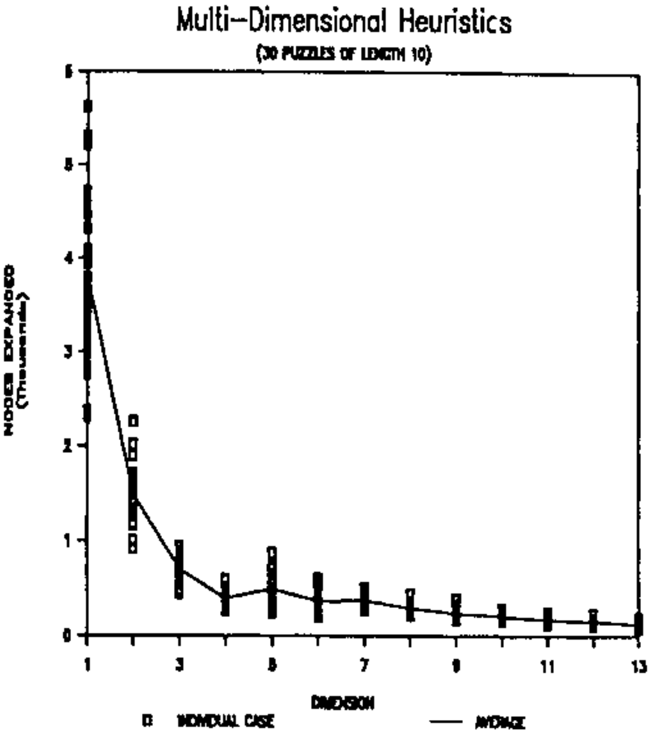

Figure 4:

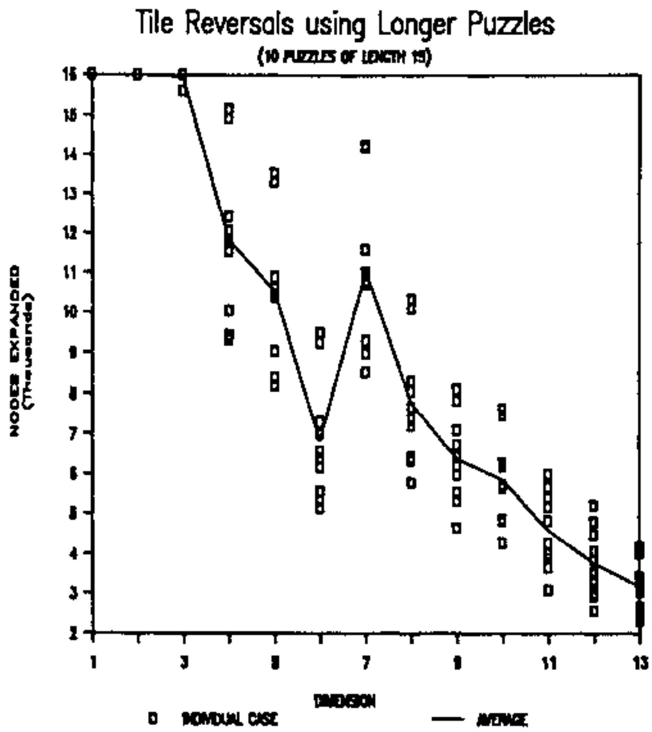

Figure 5:

by dimensions 1 or 2 because the algorithm ran out of memory. Only one puzzle was solved in dimension 3. In the higher dimensions we see a dramatic decrease in the number of nodes expanded as extra reference nodes are added. These results are shown in figure 5. Although dimension 5 is no longer a relative maxima, dimension 7 still is. Another difference in these tests is that it appears that the best value for N would be 13. Dimension 6 is a relative minima but by going out to dimension 13 the average number of nodes expanded is cut in half as opposed to using only 6 reference nodes. If the puzzle lengths are increased further the results are similar in that there is the same downward trend in the number of nodes expanded as extra reference nodes are added. As would also be expected the lowest dimension capable of solving these more difficult puzzles also increases as the path length increases.

The test results demonstrate how it may be possible to develop an MDH for a given problem domain. In this case the rather simple concept of tile reversals was by itself an inadequate heuristic. However with the addition of the multiple reference states to map the search into an N-space, the use of tile reversals as the *h* was a much better heuristic for solving 15-puzzles. This suggests that MDHs might prove especially useful for search spaces where relatively little is known about the problem. A simple heuristic *h* could be derived and a corresponding MDH might still be effective even if *h* was not.

3.2 Manhattan Distance and MDHs

An interesting question is whether MDIIs could be used to improve a good heuristic for the 15-puzzle. Some tests were run to determine the effect of MDHs when applied to the Manhattan distance heuristic. This heuristic is an admissible heuristic that is also quite good; it gives the number of moves to reach the goal if the tiles could be moved "through" each other.

The learning procedure used to find reference nodes for tile reversals proved ineffective for the Manhattan distance. Instead another "learning" procedure was used to choose reference nodes which would improve the Manhattan heuristic. Reference nodes 1 and 2 were set to be the goal and a "reversed" goal respectively. Now the 100 randomly generated nodes in [Korf, 1985] were targeted as the superset for the additional reference nodes to be added. Each of the 100 random puzzles was chosen as reference node 3 and the resulting MDH was tested on solving 10 trial puzzles. After looping through all 100 possible choices for R_3 it was found that puzzle 30 minimized the number of node expansions using 3 dimensions. Puzzle 30 was therefore chosen as R_3. This process was repeated 4 more times to pick (dimensions) reference nodes 4 through 7. This resulted in adding 5 distinct reference nodes, selected from the 100 randomly generated puzzles, which minimize node expansions when solving the 10 test puzzles. The addition of these 5 reference nodes with the original two yields an MDII with 7 dimensions. Figure 6 shows the result of solving the 10 puzzles using these reference nodes. This new MDH shows an improvement with the addition of every reference node. It also turns out that the path lengths were optimal for every puzzle solved which is surprising since this MDII is obviously not admissible.

It is a reasonable question to wonder if the improvement offered by MDHs to the Manhattan distance resulted from extra weight being placed on the heuristic. The heuristic distance estimate, h, obviously increases as more reference nodes are added, while the known

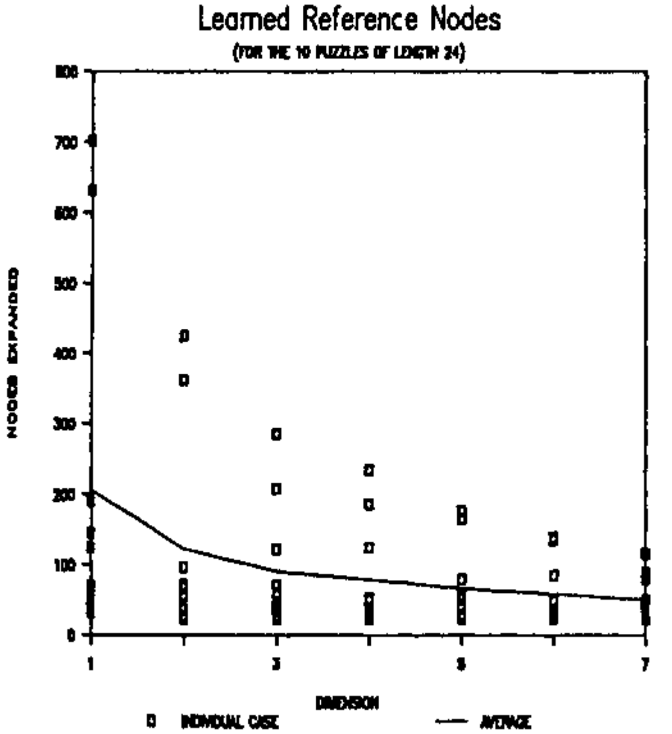

Figure 6:

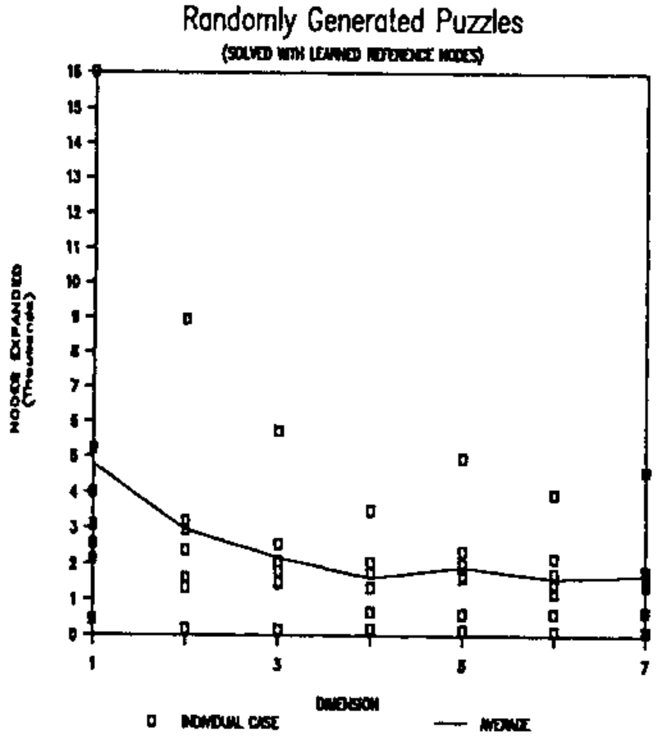

Figure 7:

distance component, g, remains the same. To explore this possibility some tests were run where all 7 reference nodes were chosen to be the goal state. The result of doing this is really just an increase in weight. There was no improvement offered by using the goal as the reference nodes. For these trials dimension 1 is the minimum or optimal MDH. It also turned out that almost half of the solutions found using these reference nodes were not optimal with respect to the length of their solution paths. Thus we can conclude that the improvement is not the result of a weight change in the heuristic.

The results in figure 6 look very promising. The set of reference nodes form an MDH which significantly improves an already good heuristic. However the manner in which the reference nodes were picked suggest the possibility that these reference nodes are "tuned" or suitable for only these 10 puzzles. After all, reference nodes 3 through 7 were chosen by a trial and error process which involved solving each of these 10 puzzles hundreds of times. If the reference nodes are only suitable for these puzzles, it is quite obvious that the overhead in choosing the reference nodes far outweighs the benefit provided by using them.

In order to test the general effectiveness of these reference nodes on other puzzles, some more tests were run using the MDH created by these 7 reference nodes to solve randomly generated 15-puzzles. Most randomly generated puzzles are not solvable by *A**, using the Manhattan distance heuristic with or without this MDH instance, because of exponential memory requirements. A "pre-screening" was conducted on these randomly generated puzzles by using the Manhattan distance to estimate a minimum path length. Any puzzle whose lower bound was greater than 20 was immediately discarded. 75 puzzles were generated with a lower bound path length < 20. Of these 75 puzzles, only 7 were able to be successfully solved by any of the 7 dimensions. The path lengths of these puzzles ranged from 22 to 30 with the average being 26. The worst case, with respect to the number of node expansions, for each of the 7 puzzles occurred in dimension 1, while the best cases were distributed over dimensions 4 through 7. For these randomly generated puzzles the MDH proved very successful. A graph of the results is shown in figure 7. There is a significant downward trend in the average number of nodes expanded as extra reference nodes or dimensions are added. So it appears that the reference nodes which were originally found for 10 specific puzzles are also capable of reducing search costs for randomly generated puzzles. In fact no randomly generated puzzle was found to expand fewer nodes in dimension 1, which is equivalent to using *A** without MDHs, than in any of the higher dimensions. Of course many puzzles were not solvable for any of the 7 dimensions. Additionally the path lengths were again optimal in every dimension for at least 6 of the 7 puzzles. The seventh puzzle was not solvable for dimension 1, the only provably admissible dimension, and therefore the optimal path length for this puzzle is not known.

The reasons for the success of MDHs in improving the 15-puzzle Manhattan distance heuristic is not as intuitive as to why MDHs improve the 15-puzzle tile reversal heuristic. Although the existence of a tile reversal between two states a and b is known to imply a significant difference between states a and b, the occurrence of a tile reversal is not likely. Using MDHs with numerous reference states in effect multiplies the likelihood of detecting tile reversals enabling the detection of previously undetected differences. When using the Manhattan distance heuristic, it is unclear that a reference state will detect any previously undetected differences. However there is a new difference that may be detected. This difference is whether or not states a and b are estimated to be equidistant from the reference state. The importance of this difference is seen in the fact that when using the Manhattan distance heuristic states a and b will be estimated to be equidistant to every state in the problem space only when a = b. Furthermore the importance of these differences is also supported by the test results which selected a good set of reference states even though, disregarding parity, only approximately $\frac{100}{20,000,000,000,000}$ or .0000000005 percent of the states in the search space were considered in selecting these reference nodes.

4 Conclusions

Heuristic searching can be improved by using a set of reference states to gain perspective on the position of a generated state with respect to the goal. Some heuristics will not detect some of the differences (distance) between two states if these states are only evaluated with respect to each other. Sometimes these differences can be found by evaluating the states with respect to a reference state and looking at the net difference between the two states and the reference state. That is, using a particular h, X_1 and X_2 may have no differences. But if h is used to compare X_1 and X_2 with a third state X_3, it may be that X_1 and X_2 do not have the same differences with X_3. In other words, $h(X_1, X_2) = 0$ implying $X_1 = X_2$, while $h(X_1, X_3) \neq h(X_2, X_3)$ showing that $X_1 \neq X_2$. For good heuristics, which will usually only yield $h(X_1, X_2) = 0$ when $X_1 = X_2$, it may seem that there is no benefit offered by introducing a third state into the evaluation process. However for any h there will be a "difference" elucidated by using a third state or reference state which is in essence a check to see whether or not X_1 and X_2 are equidistant from this third state.

When multiple reference states are used to gain perspective on the distance between X and G, a co-planar view of the search space will yield an infinite number of estimates for $\overline{XG}$. Instead the reference states are used to map the search into an N-space. This view yields only one estimate for $\overline{XG}$. Furthermore this multi-dimensional mapping has the property that if the net number of differences with respect to a reference state decreases between X and G, then so will the corresponding H_n. This is desirable because the closer two states are to each other, then the closer their difference values should be with respect to a reference state. Geometrically speaking, two points that lie near one another should be approximately the same distance from a third point. The converse of this is not always true as there exists a region of declination for every R_i and G. This region of declination can be overcome by using many reference points which are spread out. By doing so, it makes it very unlikely that any one generated state will lie in the declination regions for any significant portion of the reference nodes.

MDHs have been tested on the 15-puzzle for 2 different heuristics. MDHs were first shown to offer a significant improvement in decreasing the number of nodes expanded when a tile reversal count is used as the *h*. These tests demonstrated that a weak heuristic based on a simple concept might be greatly improved by using the multi-dimensional scheme. Furthermore once an *h* is chosen, the process of creating the MDH, that is the picking of the reference nodes, can be automated. This may prove very beneficial for a sort of "computer-aided" generation of heuristics for problems where there exist easily identifiable simple heuristic information, but for which no good heuristics are known. The second set of tests experimented with using MDHs to improve the already good Manhattan distance heuristic. MDHs were shown to be capable of improving this heuristic as well. This is somewhat surprising since the Manhattan distance heuristic is known to be one of the best 15-puzzle heuristics and shows that an MDH improvement is not exclusive to weak heuristics. An additional interesting result of the testing is that optimal solutions were found with respect to path lengths. This was unexpected since the higher dimensions of both MDH instances, particularly the Manhattan distance MDH, obviously overestimate distance values.

Further research concerning MDHs is planned in several different areas. In addition to the multi-dimensional model, there may be other models which could effectively represent, the concept of using reference states for heuristic improvement. New and improved learning procedures could be developed to pick reference states. It would also be nice to identify other problems for which MDHs may offer an improvement.

Acknowledgments

Rich Korf has been generous in allowing us to use some of his programs which included a random 15-puzzle generator.

References

[De Champeaux and Sint, 1977] D. De Champeaux and L. Sint. An Improved Bidirectional Heuristic Search Algorithm. *Journal of the ACM,* 24:177-191, 1977.

[Hart *et al*., 19(>8] P. Hart, R. Duda, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans, on SSC, 4* 1968.

[Korf, 1985] Richard E. Korf. Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artificial Intelligence,* 27:97-109, July 1985.

[Nelson, 1988] Peter C. Nelson. Parallel Bidirectional Search using Multi-Dimensional Heuristics. Ph.D. Dissertation, Dept. of Electrical Engineering and Computer Science, Northwestern University, June 1988.

[Rich, 1983] Elaine Rich. *Artificial Intelligence.* McGraw-Hill, New York, New York, 1983.