# The Pathology of Heuristic Search in the 8-puzzle

Rok Piltaver, Mitja Luštrek and Matjaž Gams

Rok Piltaver, rok.piltaver@ijs.si, +386 1 477 3230
Mitja Luštrek, mitja.lustrek@ijs.si, +386 1 477 3380
Matjaž Gams, matjaz.gams@ijs.si, +386 1 477 3644
Department of Intelligent Systems - Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

Journal of Experimental & Theoretical Artificial Intelligence, Volume 24, Issue 1, 2012
Received: 3 March 2009; Accepted: 10 October 2010; Published online: 30 June 2011

**Abstract**

In practice, an incomplete heuristic search nearly always finds better solutions if it is allowed to search deeper, i.e., expand and heuristically evaluate more nodes in the search tree. On the rare occasions when searching deeper is not beneficial, a curious phenomenon called "search pathology" occurs. In this paper we study the pathology and gain of a deeper search of the minimin algorithm in the 8-puzzle, a domain often used for evaluating single-agent search algorithms. We have analyzed the influence of various properties of the search tree and the heuristic evaluation function on the gain and the pathology. In order to investigate a broad range of the properties, the original 8-puzzle was extended with diagonal moves, yielding a larger variety of search trees. It turned out that in the 8-puzzle a substantial proportion of the solvable positions is pathological under various parameters. More importantly, the search parameters that enable the highest gains are quite consistent in pathological and non-pathological positions alike, thus pointing to potentially successful search strategies.

**Key words**: search pathology, 8-puzzle, heuristic function, incomplete heuristic search

## 1. Introduction

Many problems, such as game-playing and path-finding, can be solved by search algorithms. To do so, the problems are represented by a search graph or tree in which the nodes correspond to the states of the problem, e.g., the positions on a chess board, and the edges correspond to the actions leading from one state to another, i.e., the legal moves. Some problems have search trees that are too large to be searched completely, but they can still be solved by an incomplete heuristic search (Korf 1990): the search tree is expanded to a certain search depth, and the nodes there are evaluated using a heuristic function that assigns them heuristic values approximating to the true values. These values are backed up to the current state where they are used to select the best move. The backed-up values are usually computed using the minimax algorithm (Neumann 1928) for two-player games, such as chess, and the minimin algorithm for single-agent problems, such as path-finding and solving combinatorial puzzles. This paper focuses on the 8-puzzle, a problem from the latter group.

Practice shows that a player who considers many moves ahead usually beats an opponent who considers only a few moves in advance. However, past theoretical research showed that sometimes a deeper, incomplete search leads to worse results than a shallower search. This phenomenon is called search pathology and was discovered in the early 1980s in the case of the minimax search (Nau 1979; Beal 1980). The pathology of the single-agent search was discovered over two decades later (Bulitko, Li, Greiner and Levner 2003) and has not yet been studied as thoroughly as the pathology of the minimax search.

When setting the search depth of an incomplete heuristic algorithm one must make a compromise between the time needed for the search and the quality of decision produced by the search. Deeper search always takes more time than shallower search, while the quality does not increase with search depth when pathology occurs.

In this paper we study how much one gains by searching deeper in the 8-puzzle, a classical problem for single-agent search algorithms. A number of properties of the heuristic evaluation function are investigated to determine how they affect the gain and particularly the pathology of a deeper search. For example, we find that increasing the number of different values that a heuristic function may return tends to increase the gain and decrease the degree of pathology. The effect of the properties of the search tree on the gain is also analyzed. To this end we extend the 8-puzzle with diagonal moves, yielding several variants with varying branching factors and degrees of similarity between sibling nodes.

We also study the frequency of the pathological positions in the 8-puzzle. Although in most cases it is better, on average, to search deeper, a surprisingly large share of the positions turns out to be pathological. With the classic Manhattan-distance heuristic function, in 19.7 % of positions one is better off searching to depth 1 than to depth 5, in 31.0 % of positions a deeper search is preferable, and in 49.3 % of positions the results at both depths are equivalent. Therefore, the search in pathological positions is relevant to the overall success of the search, and more knowledge about the factors causing or preventing the pathology may lead to better search algorithms and heuristic functions.

A brief review of related work is presented in Section 2. Section 3 defines the variants of the 8-puzzle. The measures of the search pathology are described and the factors affecting the pathology are introduced in Section 4. Section 5 deals with the effect of the properties of the heuristic function on the pathology. Section 6 presents the influence of the properties of the search tree on the pathology. The frequency of the pathological positions in the 8-puzzle is discussed in Section 7. Section 8 concludes the paper.

## 2. Related work

It is well documented that a deeper heuristic search yields better solutions. For the single-agent search, this was already found in Korf's paper introducing the classic RTA* and LRTA* (Korf 1990). A comparison of several modern, real-time, single-agent search algorithms at various search depths can be found in (Hernández and Meseguer 2008). For the minimax search used in two-player games, several studies showed that a deeper search is beneficial, even though the effect of diminishing returns can usually be observed at large depths (Junghanns et al. 1997, Heinz 2001). Since the benefit of a deeper search is usually taken for granted, the papers that study factors affecting it are mostly found among the papers on search pathology.

The pathology in the minimax search was first described by Nau (1979) and Beal (1980). Since then, several attempts to explain it were published. The primary cause for the pathology in theoretical models was deemed to be the independence of the values of the terminal nodes in the search tree (Beal 1982; Bratko and Gams 1982; Nau 1983; Luštrek, Gams and Bratko 2006; Kaluža et al. 2007a, 2007b). In real games, sibling nodes in the search tree are usually similar and not independent, but the early models did not account for this. Another important reason for the pathology was found to be the low granularity of the heuristic function (Nau 1979; Scheucher and Kaindl 1998; Luštrek 2007; Kaluža et al. 2007a, 2007b). Finally, large branching factors were also observed to contribute to the pathology (Nau 1979; Luštrek et al. 2006; Kaluža et al. 2007a, 2007b).

The pathology in the single-agent search was discovered in 2003 (Bulitko 2003; Bulitko et al. 2003). In a study on synthetic search trees (Luštrek 2005, 2007), two groups of factors influencing the pathology were identified: those related to the search tree and those related to the heuristic function. In the first group, two factors were found, both involving the differences between the values of the sibling nodes in the search tree. These differences are fixed in the 8-puzzle (sibling nodes are either one move closer or one move further from the solution than their parent), so the two factors are not explored in this paper. The factors in the first group are also less interesting than those in the second group because they are given with the problem and cannot be altered. The factors in the second group are related to the heuristic function and therefore can be altered. Heuristic functions can be pessimistic, optimistic or balanced (neither optimistic nor pessimistic), and monotonically non-decreasing (the heuristic value of a node is not smaller than the heuristic value of its parent) or not. Monotonically non-decreasing functions were found to be less prone to the pathology than others (Luštrek 2005, 2007). In path-finding, the pessimistic heuristic functions were shown to be slightly less prone to the pathology than the optimistic ones (Luštrek and Bulitko 2008), and some properties of the search algorithm specific to path-finding were also shown to affect the pathology (Luštrek and Bulitko 2006).

The pathology was already observed in the 8-puzzle (Bulitko 2003, Sadikov and Bratko 2006). Experiments with seven different heuristic functions revealed three types of pathology: more non-optimal moves at a greater search depth, worse final solutions at a greater depth and fewer search-tree nodes expanded at a greater depth (Bulitko 2003). The first type is the subject of our research, while the second type is strongly correlated with it. The last type is different and unlike the first two it is actually desirable, if no less unexpected. No insights into the reasons for the pathology or the factors affecting it were given, however.

Sadikov and Bratko (2006) studied the suitability of pessimistic and optimistic heuristic functions for a real-time search in the 8-puzzle. They discovered that pessimistic functions are more suitable. They also observed the pathology, which was stronger with the pessimistic heuristic function. However, they did not study the influence of other factors on the pathology or provide any analysis of the gain of a deeper search. In our paper, the basic pathology observed in (Sadikov and Bratko 2006) was confirmed.

Recently a unifying view of the pathology in minimax and single-agent search was published (Nau et al. 2010). It showed that the dependence of the true values in the search tree, the granularity of the heuristic function and the branching factor affect the pathology in single-agent search in the same way as in minimax search. The effect of all three factors was shown on synthetic trees. The 8-puzzle was used as an example of the effect of the granularity.

This paper investigates six factors affecting the search gain and the pathology: the granularity of the heuristic function, the amount of heuristic error, early terminations, whether the heuristic function is optimistic or pessimistic, the similarity of the true values of sibling nodes in the tree, and the branching factor of the tree. Our findings agree with the work on the 8-puzzle (Sadikov and Bratko 2006) and path-finding (Luštrek and Bulitko 2008) in that pessimistic heuristic functions tend to gain more with deeper search. When analyzing optimistic and pessimistic functions we use the model of heuristic function as in (Sadikov and Bratko 2006) and path-finding (Luštrek and Bulitko 2008). We confirm the effect of the dependence of the true values in the search tree on the pathology, which was so far only shown on synthetic trees in single-agent search (Nau et al, 2010). We also provide an explanation for the effect of the granularity. The remaining factors, most importantly early terminations, were not studied in the past. Furthermore, we also examine the gain in addition to the pathology and the influence of various search depths (not just 1 and 5).

## 3. The 8-puzzle and its variants

The 8-puzzle is a sliding puzzle that consists of a grid of numbered tiles with one tile missing. If the grid is n × n, the puzzle is called the (n2 – 1)-puzzle or the n2-puzzle (Ratner and Warmuth 1990; Ryan 2008). The numbers on the tiles are initially in a random order. The goal of the puzzle is to rearrange the tiles in as few moves as possible so that the numbers are in order. The moves are made by sliding a tile into the empty space, in turn revealing another empty space in the position of the moved tile (Wikipedia 2009). An example of solving the 8-puzzle is presented in Figure 1. An initial state is shown in Figure 1a, followed by consecutive states, move by move (b, c, d), until the puzzle is solved in Figure 1e.

| | | |
|---|---|---|
| | 2 | 3 |
| 1 | 4 | 6 |
| 7 | 5 | 8 |

a)

| | | |
|---|---|---|
| 1 | 2 | 3 |
| | 4 | 6 |
| 7 | 5 | 8 |

b)

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | | 6 |
| 7 | 5 | 8 |

c)

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | | 8 |

d)

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | |

e)

*Figure 1. An example of solving the 8-puzzle*

The n2-puzzle is a classical problem that is suitable for the study of search algorithms (Korf 1990; Reinefeld 1993) – its state space is easily defined with a few simple rules and there is no known algorithm for solving the puzzle efficiently. Finding a solution for a puzzle of an arbitrary size is easy, but the problem of finding the shortest solution is NP-hard (Ratner and Warmuth 1986). In our experiments, the 8-puzzle was chosen over larger n-puzzles because the size of its state space is small enough for extensive analyses. There are 9! / 2 = 181,440 solvable start positions (Ryan 2008) and it takes, at most, 31 moves to solve any of them. Therefore, it is possible to find the optimal solutions for all the solvable start positions in a matter of seconds on a personal computer.

The state space of the 8-puzzle also has some specific properties. Firstly, in every position that can be solved optimally in $n$ moves, one can make only moves that will lead to positions requiring $n – 1$ or $n + 1$ moves to solve them. In other words, each move decreases or increases the minimum distance to the solution by exactly one move. As a consequence, there are no neutral moves, all moves in the direction towards the solution are equally good and all moves away from the solution are equally bad. Secondly, the number of positions varies with the distance to the solution. Only 1.5 % of all the positions are less than 14 moves away from the solution. Over 90 % of the positions are between 17 and 27 moves away from the solution. The average distance to the solution is 21.9 moves. This is presented in Figure 2, where gray bars show the number of positions for each possible distance to the solution. Thirdly, the percentage of positions in which all possible moves are optimal increases with the distance to the solution. The number of positions in which all possible moves are optimal is shown with black bars in Figure 2.

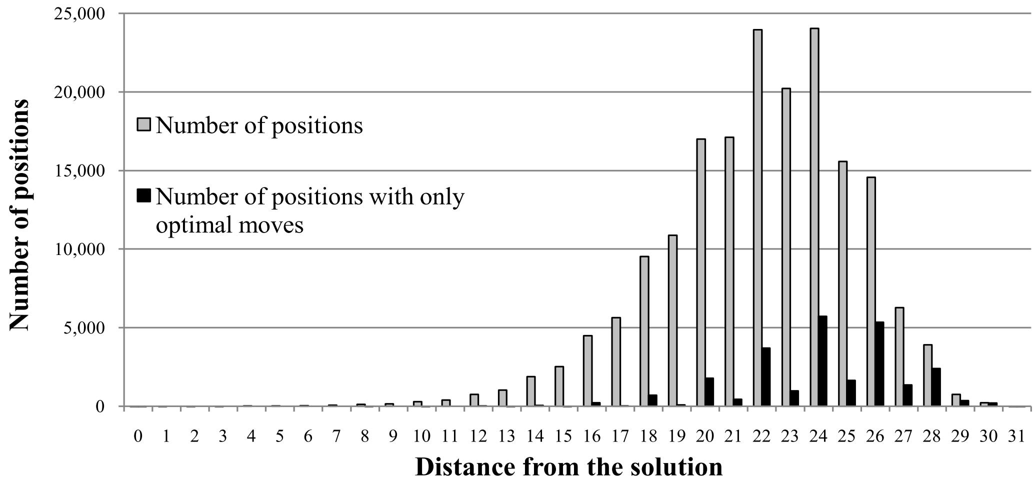



*Figure 2. The number of positions and the number of positions with only optimal moves for a given distance to the solution in the usual 8-puzzle*

To measure the influence of the branching factor and the similarity of sibling nodes on the pathology, variants of the 8-puzzle were introduced. The branching factor of the search tree can be increased by adding additional moves to the original puzzle, or decreased by disallowing certain moves. Besides sliding a tile into the empty slot above, below, left or right, we allowed switching a tile with an empty slot diagonally from it. Figure 3 illustrates the 8 possible directions of the move, e.g., up & left (UL). A specific variant of the 8-puzzle is defined by allowing $k$ of these 8 possible moves (4 basic and 4 diagonal) and disallowing the other $8 - k$ moves. There are 255 such variants of the 8-puzzle, of which 129 are not interesting as they have only a small number (< 202) of solvable start positions – they are disregarded in this paper. Of the remaining 126 variants, 31 of them, including the usual 8-puzzle, have 9! / 2 solvable start positions, while 95 variants have 9! solvable start positions. Even though these variants were designed to vary in the branching factor, they are different in other respects as well. Among them is the degree of similarity between the sibling nodes, the average number of moves required to solve a random start position, and other properties.

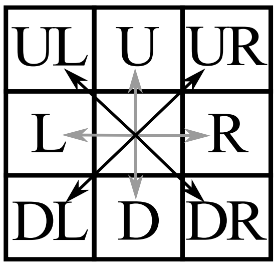



*Figure 3. Four basic (gray) and four additional (black) directions of the move*

## 4. Problem formulation

The problem of solving the 8-puzzle is represented by a search tree in which the nodes correspond to the positions of the tiles in the puzzle and the edges correspond to the legal moves leading from one position to another. In order to select the best move in a position $m$, the tree rooted in $m$ is expanded to a depth $d$, as shown in Figure 5. The evaluation function assigns each position $p$ at the depth $d$ a heuristic value $h(p)$. This value approximates the true value $h^*(p)$, which equals the minimum number of moves needed to reach the goal position (i.e., to solve the puzzle) from the position $p$. The heuristic values at the depth $d$ are then backed up to level 1, which consists of the positions one move away from the root *m*. This is done using the minimin algorithm, which recursively computes the value of each position *o* as the minimum of the values of its descendants, increased by 1. The rationale for this rule is that in the position *o*, one should choose the move leading to the descendant *do* that is closest to the solution; however, since *o* is one move away from *do*, the value of *o* should be larger by 1 than the value of *do*. The minimin algorithm thus assigns the backed-up heuristic value $h_{d-1}(n)$ to each position *n*, one move away from the root *m*. This value is equal to the minimum of the heuristic values of the leaves of the tree with the depth $d - 1$ rooted in *n*, increased by $d - 1$. Finally, the move selected in the position *m* after searching *d* moves ahead is one that leads to one of *m*'s descendants with the lowest backed-up heuristic value $h_{d-1}$ (shown on a gray background in Figure 5). If the goal position is encountered at the depth $d_g < d$, the backed-up heuristic value is $d_g - 1$. Such a situation is called an *early termination*.

The heuristic functions used is of the form $f = d + h$, where *h* represents the heuristic estimate of the position (Russel, S, Norvig, P 2005), in our case the number of the moves to the goal position. Since *d* is an integer representing the search depth to the particular position under estimation, the term heuristic function is often referred to *h*.

**Definition 1**: The probability of a correct (optimal) move in a position *m* after searching *d* moves ahead is calculated using Equation (1). This probability is 1 – probability of a move error, denoted as $wrong_d(m)$.

$$correct_d(m) = \frac{\# n \in Desc(m) : \left( h_{d-1}(n) = \min_{o \in Desc(m)} \{h_{d-1}(o)\} \right) \wedge \left( h^*(n) = \min_{o \in Desc(m)} \{h^*(o)\} \right)}{\# n \in Desc(m) : h_{d-1}(n) = \min_{o \in Desc(m)} \{h_{d-1}(o)\}} \quad (1)$$

$$\text{wrong}_d(m) = 1 - \text{correct}_d(m) \quad (2)$$

*Desc*(*m*) is the set of the descendants of the node *m*, $h_d(n)$ is the backed-up heuristic value of a node *n* obtained by searching *d* moves ahead (when $d = 0$, the index is omitted), and $h^*(n)$ is the true value of the node *n*. The move selected at the node *m* leads to one of its descendants with the lowest backed-up heuristic value. The number of such nodes is in the denominator of the right-hand side of Equation (1).

In the case of multiple nodes with the same minimum heuristic value, one is chosen at random. The chosen move is correct if the true value of the node it leads to is minimum among its siblings. The number of such nodes is in the numerator. In short, the probability of a correct move at the node *m* equals the probability of randomly selecting from the descendants of *m* with the lowest heuristic value the one with the lowest true value.

Examples of calculating $correct_1(m)$ and $correct_5(m)$ are shown in Figures 4 and 5, respectively. The nodes with the gray background represent the best nodes at a given level of the search tree according to the heuristic function (note that the values at depths 1 and 5 differ by four because four more moves are needed to the goal from depth 1). Out of these, the nodes marked with rectangles are the truly optimal ones, whereas those marked with circles are wrongly estimated as optimal.

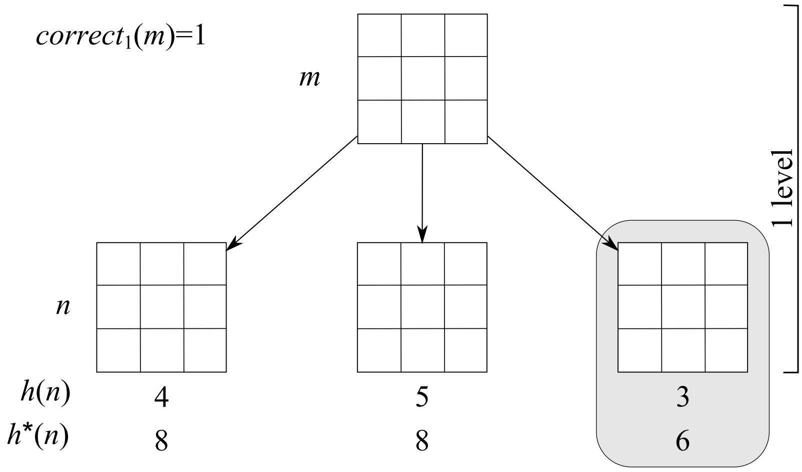



*Figure 4. An example of calculating $correct_1(m)$*

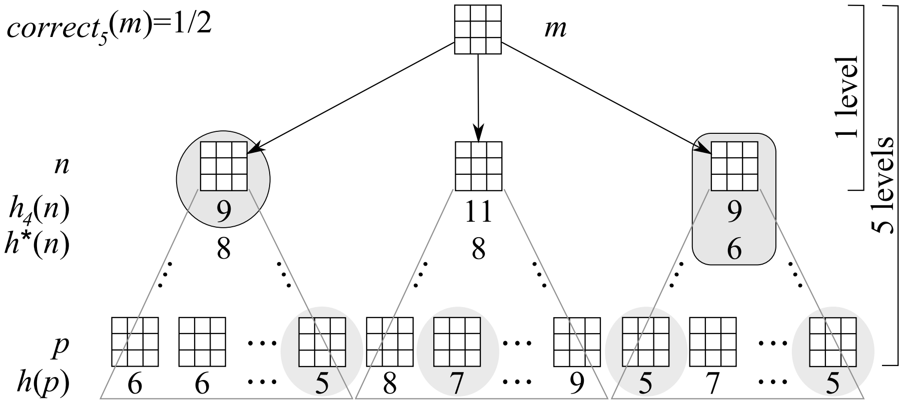



*Figure 5. An example of searching to depth 5 and calculating $correct_5(m)$*

**Definition 2**: A position *m* is *pathological* for depths *i* and *j* iff $correct_i(m) < correct_j(m)$ for $i > j$, and *non-pathological* otherwise.

For the example in Figures 4 and 5, $correct_5(m) = ½$ and $correct_1(m) = 1$ and $5 > 1$, therefore the position *m* is pathological for depths 5 and 1.

The probability of a correct move can be measured either for a single position, for a group of positions or for whole domains, e.g., the variants of the 8-puzzle.

**Definition 3**: The probability of a correct move in an 8-puzzle variant *M* when searching *d* moves ahead is calculated using Equation (3). It equals the average probability of a correct move over all the solvable positions in the puzzle.

$$avrCorrect_d(M) = \frac{\sum_{i \in SolvPos(M)} correct_d(i)}{|SolvPos(M)|} \tag{3}$$

The probability of a wrong move is correspondingly:

$$avrWrong_d(M) = \frac{\sum_{i \in SolvPos(M)} wrong_d(i)}{|SolvPos(M)|} = \frac{\sum_{i \in SolvPos(M)} 1 - correct_d(i)}{|SolvPos(M)|} \tag{4}$$

*SolvPos*(*M*) is the set of all the solvable positions of the 8-puzzle variant *M*.

**Definition 4**: The *search gain* (or *gain* for short) of an 8-puzzle variant *M* is defined as the ratio between the probability of a correct move in *M* when searching to depths *i* and *j*, where $i > j$. It is calculated using Equation (5).

$$gain_{i,j}(M) = \frac{avrCorrect_i(M)}{avrCorrect_j(M)} = \frac{1 - avrWrong_i(M)}{1 - avrWrong_j(M)} \quad (5)$$

A deeper search (to depth *i* instead of *j*) is worthwhile iff $gain_{i,j}(M) > 1$. Iff $gain_{i,j}(M) = 1$, the average quality of the moves is the same when searching to depths *i* or *j*, so depth *j* is preferable because it saves time.

**Definition 5**: The *degree of pathology* (or *pathology* for short) of an 8-puzzle variant *M* is defined as the ratio between the probability of a wrong move in *M* when searching to depths *i* and *j*, where $i > j$. It is calculated using Equation (6).

$$pat_{i,j}(M) = \frac{1 - avrCorrect_i(M)}{1 - avrCorrect_j(M)} = \frac{avrWrong_i(M)}{avrWrong_j(M)} \quad (6)$$

The degree of pathology corresponds to the ratio between the probability of choosing a wrong first move if searched to depth *i* and *j*, while the gain corresponds to the ratio between the probabilities of choosing a correct first move, and as a consequence the degree of pathology and the gain are not inversely proportional, although in general a lower gain corresponds to a higher degree of pathology. For example, compare $pat_{5,1}(n) = 1.001$ and $gain_{5,1}(n) = 0.5$ resulting from $avrCorrect_5(n) = 0.001$ and $avrCorrect_1(n) = 0.002$ on the one hand, and $pat_{5,1}(m) = 1.1$ and $gain_{5,1}(m) = 0.6$ resulting from $avrCorrect_5(m) = 0.12$ and $avrCorrect_1(m) = 0.2$ on the other hand. Both the degree of pathology and the gain increase from *n* to *m*. On the other hand, iff $gain_{i,j}(m) > 1$, then $pat_{i,j}(m) < 1$ and vice versa.

**Definition 6**: A domain *M* is pathological for depths *i* and *j* iff $gain_{i,j}(M) < 1$, i.e. $pat_{i,j}(M) > 1$.

The results in this paper are mainly presented in terms of $gain_{5,1}$ and $pat_{5,1}$ One of the reasons for choosing these depths is that the studies in minimax were often performed at depths 5 and 1; therefore the same depth in the single-agent search makes possible a better comparison. The other reason is that these depths enable studies without the effect of early terminations. Other search depths, such as 6 and 1 or 8 and 1, were also analyzed, providing qualitatively very similar relations. A detailed analysis of the influence of the various search depths *i*, *j* is presented in Sections 5.1 and 5.2.

In Sections 5 and 6, the following issues related to the pathology and gain are studied:

- The granularity of the heuristic function, i.e., how many different values it can return (Section 5.1).
- The granularity of the heuristic function, premature terminations or not and always included noise or not (Section 5.2).
- Fluctuations at small granularities (Section 5.3).
- Tentative explanation of the influence of granularity (Section 5.4).
- The amount of heuristic error (Section 5.5).
- Different heuristic functions: optimistic, pessimistic and two real-life (Section 5.6).
- The similarity of the true values of the sibling nodes in the tree (Section 6.1).
- The branching factor of the search tree, i.e., the number of moves at each node in the tree (Section 6.2).

## 5. Influence of the heuristic function on the pathology and gain

**Definition 7**: In the 8-puzzle, the *true value* or *quality* of a given position is equal to the minimum number of moves needed to solve the puzzle from that position. The difference between the *heuristic value* $h(n)$ and the true value $h^*(n)$ of a position $n$ is called the *heuristic error* $e$. The heuristic value is computed using the *heuristic evaluation function* (or heuristic function for short) $h$.

The error of the *generic heuristic function* was modeled using Gaussian noise (Box and Muller 1958). The term "generic" is used because by adjusting the noise distribution, the function is intended to approximate any heuristic function that can be used in practice. The basic version of the generic heuristic function returns true values corrupted by the Gaussian noise with the standard deviation $\sigma = 2.5$, which is the same as the standard deviation of the Manhattan-distance heuristic function. The latter is often used for solving $n^2$-puzzles and is described in Section 5.6. The true values for the positions $\{n\, ;\, h^*(n) \leq 7\}$ are not corrupted because there are so few such positions that it is practically impossible to corrupt their values and maintain a constant dispersion (Sadikov and Bratko 2006) and because the Manhattan heuristic function estimates at these depths without errors. Throughout the paper this generic heuristic function is used, unless written otherwise in specific sections.

The true values $h^*(n)$ were computed for all the positions of the 8-puzzle and its variants using the retrograde analysis (Thompson 1986). Afterwards, the heuristic values $h(n)$ were computed, in the case of the generic heuristic function by adding the Gaussian noise to the true values. Finally, the backed-up heuristic values $h_i(n)$ resulting from searching $i + 1$ moves ahead were obtained using the minimin algorithm. The values $h^*(n)$, $h(n)$ and $h_4(n)$ are needed to compute $gain_{5,1}$ and $pat_{5,1}$, which are the measure of the search gain and pathology used in the experiments in this paper – see Equations (1) through (6). For other values $i$ and $j$, the corresponding $h_j(n)$ and $h_i(n)$ are computed and stored, making it possible to compute the gain and the degree of pathology.

### 5.1 Granularity of the heuristic function

The values of a heuristic function in our experiments were either real or integer numbers, or a finite subset of either of them.

**Definition 8**: The cardinality of the range $B$ of a heuristic function $h$ is called the *granularity* of the heuristic function and is denoted $g(h)$ – see Equation (7).

$$g(h) = |A|; h : A \rightarrow B \tag{7}$$

The range of the generic heuristic function is real numbers. In order to obtain a heuristic function with the desired granularity $g$, the number of possible heuristic values is reduced by creating $g$ intervals of equal size. Firstly, the heuristic values are limited to the $[0, M]$ interval, where $M = \max_n\{h^*(n)\} + \lfloor\sigma\rfloor + 1$. The value of $M$ is set close to the maximum unrestricted heuristic value to ensure that few such heuristic values are greater than $M$. If an unrestricted value $> M$, it is set to $M$, and if it is $< 0$, it is set to 0. Then, all the heuristic values are multiplied by $g / M$, to scale the interval of possible heuristic values to $[0, g]$, and finally rounded to the closest integer value.

The relation between the granularity of the generic heuristic function and the gain is presented in Figure 6, and the relation between the granularity and the degree of pathology is presented in Figure 7. The values on the horizontal axis represent the granularity of the heuristic function. The values increase linearly from 2 to 46 and exponentially from 64 to 2048. The values on the vertical axis represent the gain $gain_{5,1}$ in Figure 6 and the degree of pathology $pat_{5,1}$ in Figure 7. In Figures 6 and 7, the gain and the degree of pathology are plotted for the usual 8-puzzle and for the variant with minimum (R, D, DL, L, UL) and maximum (all directions) pathology and minimum (all directions) and maximum (R, DR, D, DL, L, UR) gain. The amount of pathology or gain over different granularities is calculated as a sum over $g$ = 2, 3, 4, ..., 46; 64, 128, 256, 512, 1024, 2048.

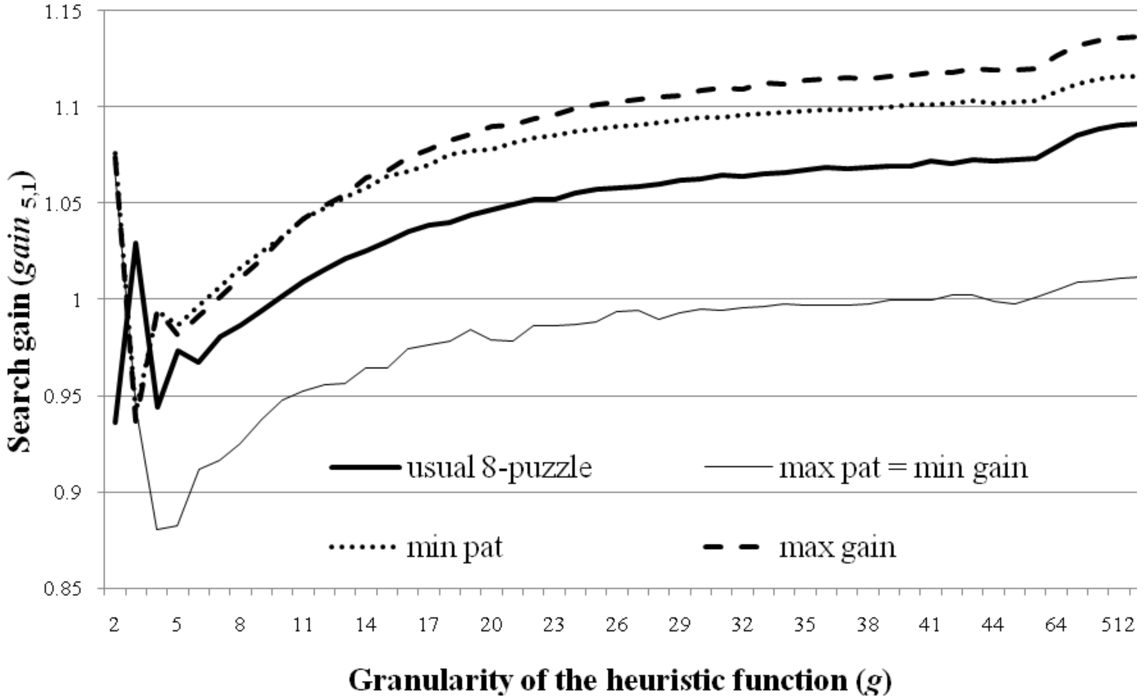



*Figure 6. The influence of the granularity on the search gain using the generic heuristic function for four variants of the 8-puzzle*

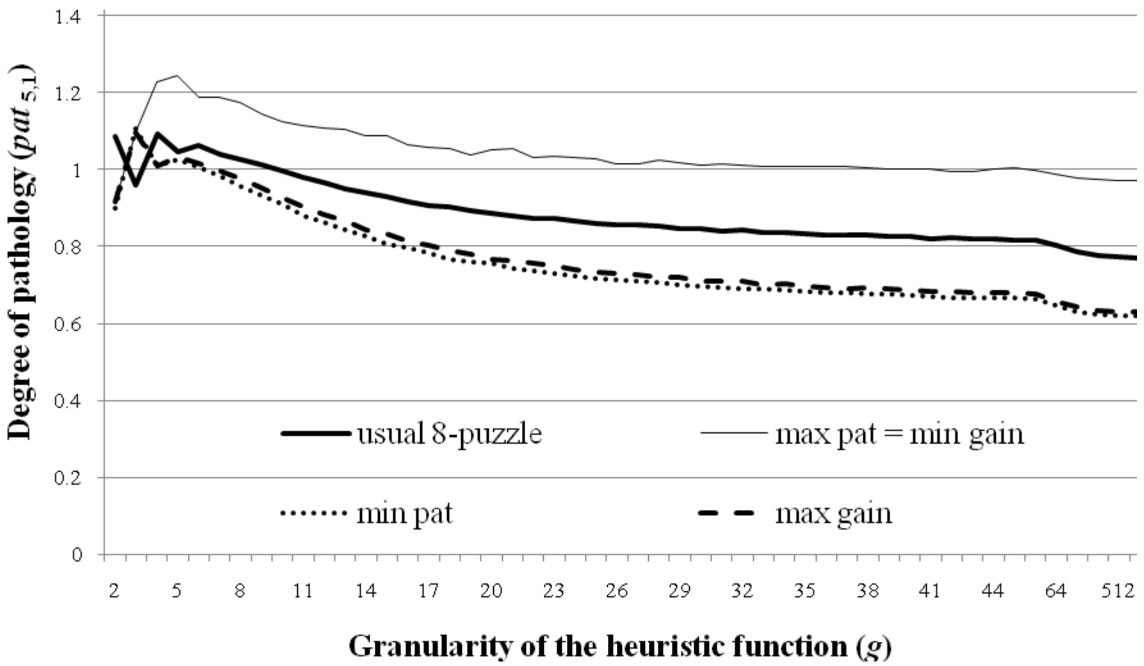



*Figure 7. The influence of the granularity on the degree of pathology using the generic heuristic function for the four variants of the 8-puzzle*

Figure 6 indicates that searching deeper is beneficial in most of the domains under granularities > 15. The exception is at small granularities, especially up to 10. The worst domain for gains is the variant with moves in all directions, where deeper search is not beneficial for a granularity < 40. Figure 6 also indicates that with the exception of granularities smaller than 15, most of the domains are not pathological.

Figure 7 shows approximately symmetric relation of the degree of pathology compared to the gain, presented in Figure 6. For granularities greater than 10, the worst pathology is close to 1.1, while the worst gain is 0.95. At granularity 1024, the degree of pathology is inside the interval (0.6 .. 1) and the gain inside (1 .. 1.15). The pathology can be found in any version of the 8-puzzle if the generic heuristic function with a small granularity is used.

In another observation, Figures 6 and 7 show that except for a granularity of 10 or less, the heuristic functions with higher granularity enable bigger search gains or – in other words – are less prone to the pathology. A qualitatively similar relation between the degree of pathology and the granularity was observed and explained in the minimax pathology (Nau 1979; Scheucher and Kaindl 1998; Kaluža, Luštrek and Gams 2007a; Kaluža, Luštrek, Gams and Tavčar 2007b; Luštrek 2007; Piltaver, Luštrek and Gams 2007), but our work is the first to show it in the single-agent search as well.

Qualitatively, the same relation was also observed in the variants of the 8-puzzle not presented in Figures 6 and 7. When the degree of pathology was approximated using a linear function for each of the following intervals $g \in [10, 20]$, $g \in [20, 30]$, $g \in [30, 40]$, $g \in [40, 50]$ and $g \in \{64, 128, 256, 512, 1024, 2048\}$ they all had negative slopes, thus confirming the relation between the search gain and the granularity of the heuristic function for all 8-puzzle variants. The calculated slopes also show that the pathology is decreasing with increasing granularity faster for low (e.g. $g = 15$) granularities than for high granularities (e.g. $g = 256$). This leads to the first observation:

**Observation 1:** From Figures 6 and 7 and the additional calculations described in the previous paragraph it follows that for the generic heuristic function, the search gain ($gain_{5,1}(n)$) is increasing with the increasing number of possible heuristic values (granularity) if $g > 10$ for all 8-puzzle variants with the exception of the smaller fluctuations. Correspondingly, the trend of the degree of pathology is decreasing with increasing granularity.

There are a maximum of 47 true values for any version of the 8-puzzle (32 for the usual 8-puzzle), so it may seem surprising that there are still changes in the pathology and gain at granularities above this number. This is possible because higher granularities of heuristic values can still affect the error by revealing the direction in which a position is misevaluated. For example, let us choose between the positions with true distances to the goal 15 and 17. If the granularity is low and their heuristic values are 16 and 16, one has to choose randomly. If the granularity is high and their heuristic values are 15.6 and 16.4, the search is able to correctly choose the first position.

To further analyze the stability of the observed relations, the influence of the search depth on the move error for heuristic functions with various granularities in the usual 8-puzzle was studied. Figure 8 shows the move errors at various search depths, from which the degree of pathology $pat_{i,j}$ can be computed for any pair of depths $i$ and $j$ by simply comparing the errors at those depths – see Equations (5) and (6). Note that the curves in the figures are smoothed in relation to the search depth which is always an integer number. This figure graphically shows that $pat_{i,j}$ for $i$ and $j$ other than 5 and 1 can be expected to behave similarly to $pat_{5,1}$ for values up to approximately 15 (note that there are a couple of exceptions). This conclusion follows from the curves for granularities 5 and more being roughly monotonic: for $g = 5$ and $g = 7$ the error mostly increases (pathology in Figure 7), for $g = 11$ the decision is not clear-cut (degree of pathology near 1 in Figure 7), and for $g$ = 1024 the error mostly decreases (no pathology in Figure 7). Furthermore, the oscillating move error of heuristic functions with a small granularity is consistent with the non-monotonic behavior of the degree of pathology and the gain at small granularities as observed in Figures 6 and 7. The sharp decrease in move error at high search depths is the consequence of many start positions being solved in less than $d$ moves, i.e., early terminations.

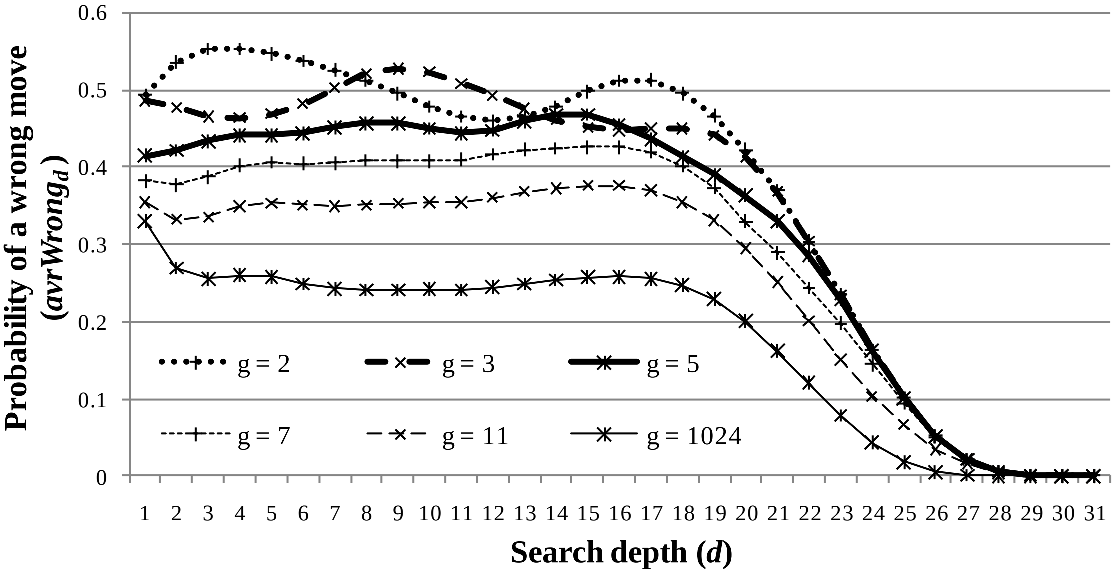



*Figure 8. The move error with respect to the search depth for generic heuristic functions with various granularities in the usual 8-puzzle*

The relations discussed in Figure 8 can be presented in a 3D view. In Figure 9, the granularities are presented as one axis, the other two axes relate to *avrCorrect* instead of *avrWrong* in Figure 8, and the search depth. Figure 8 graphically indicates that $gain_{5,1}$ behaves similarly to the gains at other depths up to 15. The relation between the depth and *avrWrong* and *avrCorrect* in Figures 8 and 9 can graphically be approximated by two lines: the first one is horizontal for depths from 2 to 15, and the second line reaches the extreme possible value at the depth 27. Together with analyses of the degree of pathology performed, here omitted due to the lack of space, this makes possible another observation:

**Observation 2**: The relation between the granularity and $gain_{5,1}$ (or $pat_{5,1}$) is similar to the relation between the granularity and the gain (or the degree of pathology) at other depths *i*,*j* smaller than 15.

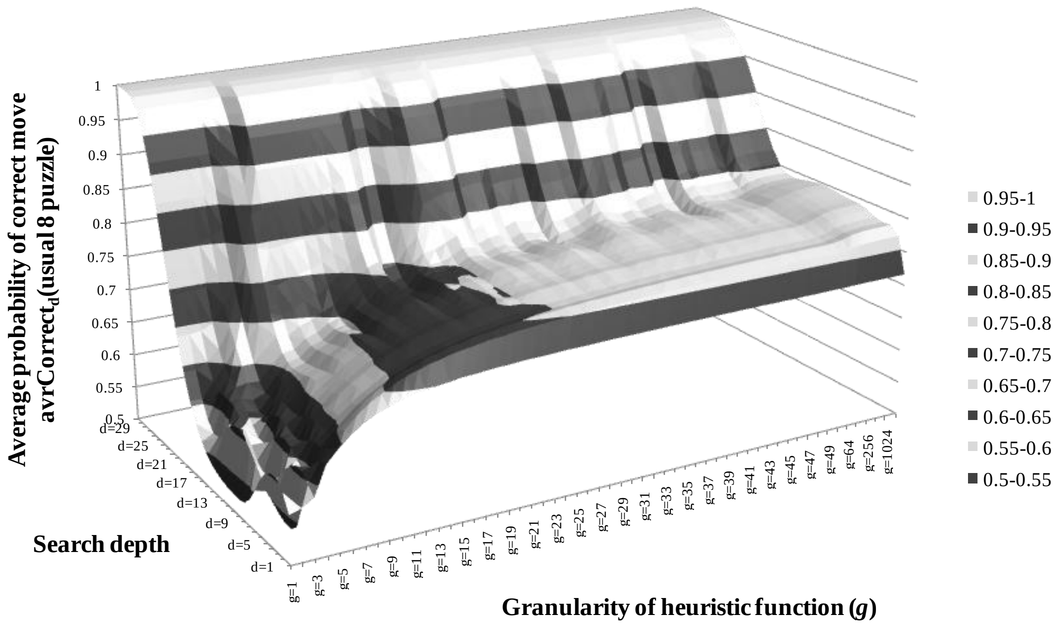



*Figure 9. The probability of correct move with respect to the search depth for generic heuristic functions with various granularities in the usual 8-puzzle*

From the $avrWrong_d$ and $avrCorrect_d$ presented in Figures 8 and 9, the gain and the degree of pathology for the depths *i* and *j* can be computed. Figure 10 shows the degree of pathology at the depths *i* and *j*, plotted at two granularities: $g = 10$ (the lowest *g* at which the pathology no longer fluctuates) and $g = 2048$ (the maximum *g* tested) for the usual 8-puzzle. Note that the graphical representation shows actual experimental results only for $i > j$; the top-left plane where $i < j$, and the middle-right "curtain" at $i = j$ are there only to make the figure visually transparent.

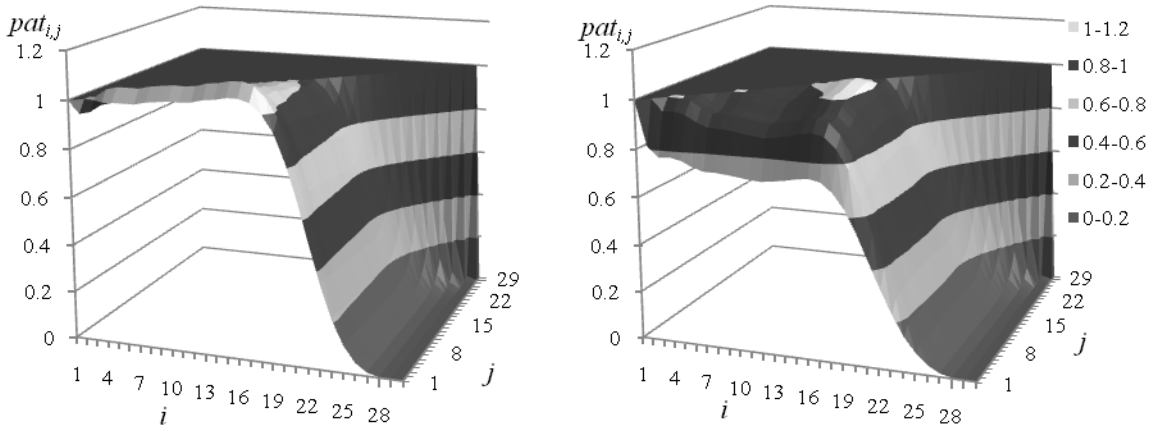
We also verified how the choice of *i* and *j* affects $\text{pat}_{i,j}$ by linearly approximating $\text{pat}_{i,j}$ with respect to *g* for $i \in [2, 15]$, $j < i$, and $g \in [10, 20]$, $g \in [20, 30]$, $g \in [30, 40]$ and $g \in [40, 50]$. We omitted $g < 10$ due to fluctuations of the error at small *g* (discussed in Section 5.3). For $g \in [10, 20]$, $\text{pat}_{i,j}$ decreased with respect to *g* in all cases when $j < i - 3$ as well as in most other cases. For $g \in [20, 30]$, $\text{pat}_{i,j}$ decreased with respect to *g* in all cases (although the decrease was sometimes small). For larger *g*, $\text{pat}_{i,j}$ decreased with respect to *g* in cases when *i* was small and when the difference between *i* and *j* was large. In other cases, it sometimes slightly decreased and sometimes slightly increased, which means that once *g* is large enough, further increases in *g* have little effect on $\text{pat}_{i,j}$.

Figure 10 reveals that most of the area with search depths up to 20 is pathological for $g = 10$, with $pat_{i,j} < 1.1$. For $g = 2048$, only small portions of the search depth pairs remain pathological, among them the biggest is inside the triangle [(12,12), (5,17), (17,17)]. Analyses at other granularities (*g* from 2 to 2048) reveal observations 3 and 4:

**Observation 3**: Regardless of the granularity *g*, for the usual 8-puzzle and the generic heuristic function, there is always some area where the degree of pathology $pat_{i,j} > 1$.

**Observation 4**: Regardless of the granularity *g*, for the usual 8-puzzle and the generic heuristic function, the $pat_{i,j}$-surface can be approximated by two separated parts. For $pat_{i,j}$ with $i \leq 19$, the surface is fluctuating for $g < 10$ (only $g = 10$ is presented here), but it is close to 1. For $i > 19$, $pat_{i,j}$ decreases in the direction of 0. In this regard, $pat_{5,1}$ is reasonably similar to most of $pat_{i,j}$ for *i* and *j* smaller than 15.

Observation 4 is similar to Observation 2, but is based on a different viewpoint.




a) $g$ = 10 b) $g$ = 2048

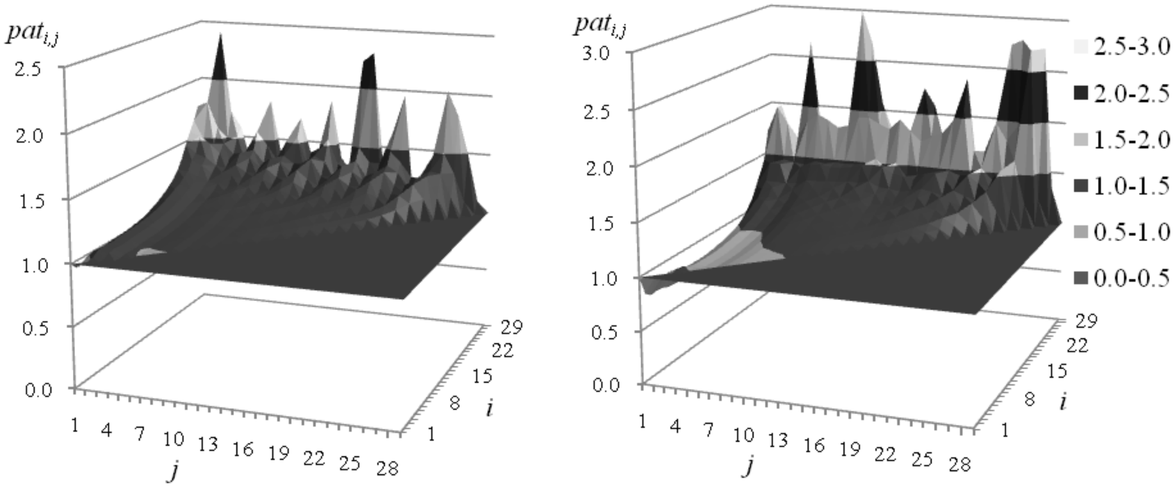
*Figure 10. The degree of pathology $pat_{i,j}$ at the granularities 10 (a, left) and 2048 (b, right) for the generic heuristic function in the usual 8-puzzle*

### 5.2 Search without perfectly estimated positions in relation to the granularity

Until now, the search trees included early terminations and positions with perfect heuristic estimations, which might both serve as anchors stabilizing search. In this section we deal with the search, where all the positions encountered include some noise. To study the granularity vs. pathology relation, the degree of pathology at depths $i$ and $j$ was computed and plotted at a specific $g$, once with (Figure 10) and once without early terminations and an exact evaluation for the positions up to 7 moves to the goal (Figure 11).

Figure 11, therefore, presents two additions compared to Figure 10. Firstly, early terminations are discarded, meaning if a goal position is found during a search, this search is discarded. After this modification only, the obtained figures were similar to Figure 10, just the turning-point at $i$ = 19 moved to 22. The second modification was also introducing noise to the error-free positions with 7 moves or less to the goal position (for an explanation, see Section 5). Since there is only a small number of these positions (see Figure 2), the tests were repeated 10 times and averaged to obtain up to a point stable results. Still, the spikes at the top-right end of the graphs in Figure 11 indicate that exact numbers should not be trusted too much due to instability.

Under the two additions to the original search model, the search becomes predominantly pathological as in the early academic models of minimax (Beal 1982; Bratko and Gams 1982; Nau 1983) or in the P-game (Pearl 1984). The major difference is that our analysis is dealing with a realistic one-player game, and not an academic two-player game. To our best knowledge, this is the first case that any real-life game was shown to be pathological under nearly reasonable assumptions – no perfectly estimated positions found in the search and the heuristic function with the Gaussian noise. While it often happens that the search finds no perfectly estimated position, the heuristic function remains an academic one, being better than random noise and worse than the Manhattan-distance function. However, as Figure 11 reveals and as was also found in the P-game and in the early academic minimax models, smaller heuristic error in a pathological search often makes search worse. After all, random search by definition results in $pat_{i,j}$ = 1.




a) $g$ = 10 b) $g$ = 2048

*Figure 11. The degree of pathology $pat_{i,j}$ at the granularities 10 (a) and 2048 (b) for the generic heuristic function in the usual 8-puzzle; no early terminations, heuristic values of positions up to 7 moves to the goal corrupted with the noise*

**Observation 5**: For the usual 8-puzzle, in the case of avoiding early terminations and when the generic heuristic function of the positions up to seven moves to the goal is also corrupted with the Gaussian noise, the search is mainly pathological and generally more pathological with a higher $i$ at fixed $j$ regardless of the granularity.

The phenomenon that positions with small or zero heuristic error named "stabilizing seeds" (Bratko and Gams 1982) or "traps" (Pearl 1983) eliminate pathology on its own, if common enough, was proposed nearly 30 years ago for the minimax search. The analyses of the 8-puzzle indicate that these positions might be the most important single factor for combating pathology in single-player games as well.

The pathology at specific search depths in the 8-puzzle was briefly observed in (Sadikov and Bratko 2006), but without in-depth analyses of factors influencing it, e.g., presenting overall pathology under the two explicit modifications of the original search model.

### 5.3 Explanation of the fluctuations at small granularities

The degree of pathology and the gain when analyzed with the generic heuristic function fluctuate at a smaller granularity, as first observed in Figures 6 and 7. Here we provide a tentative explanation of this phenomenon by analyzing the heuristic values of positions without added noise at a chosen granularity, hence the heuristic error stems only from the introduction of the granularity. Consider the example in Figure 12 for granularity 2, which transforms the true values of the positions into two values of the heuristic function: 1 for the true values from 1 to 16 and 2 for the values from 17 to 31. Let us consider a position which is 16 moves away from the solution. Its descendent at depth 1 with value 15 will become expanded into leaves at depth 5 with values greater than or equal to 11, while the sibling position at depth 1 with value 17 will become expanded into leaves with values greater than or equal to 13. At granularity 2, the true values of the two sibling positions at depth 1 will be transformed into 1 and 2, while their best descendants at depth 5 will both be 1. This will render a deeper search uninformed and hence pathological. But there are other combinations of positions, for example, true values (20, 22) at depth 1, where the search at depth 1 is completely uninformed, while at depth 5 it always finds the correct move, as presented in Figure 12.

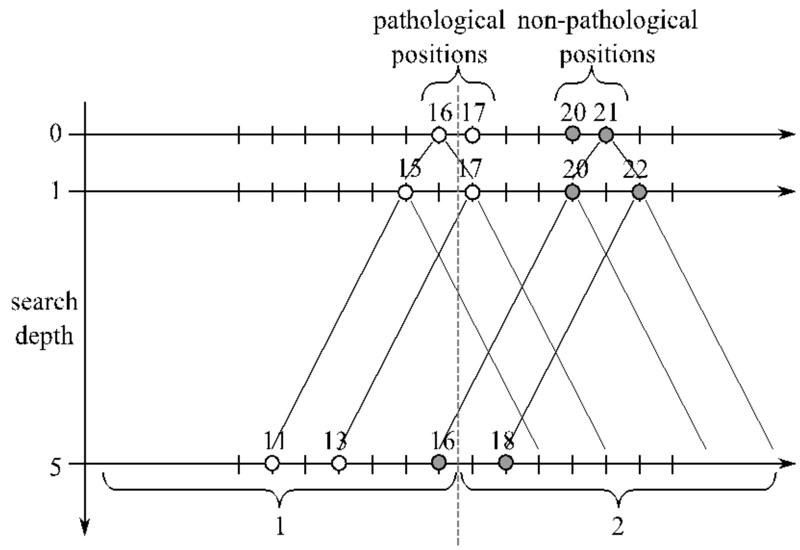



*Figure 12. Examples of pathological and non-pathological positions at the granularity 2*

To calculate an average gain and the degree of pathology at a chosen granularity, all the positions with values from 0 to 31 and with frequencies presented in Figure 2 were examined using the following procedure:

```
wrongDecisionsL1 = 0
wrongDecisionsL5 = 0
for depth = 5 to 30  begin {in reality we calculate from 0 to
31, but for the border cases additional tests of borders are
needed, making the code a little more complex}
  correctMoveL1 = depth - 1
  wrongMoveL1 = depth + 1
  if (gr(correctMoveL1) == gr(wrongMoveL1)) wrongDecisionsL1+=
    numOfPosAtDepth[depth] * (1 - averageCorrect[depth])
  correctMoveL5 = depth - 5
  wrongMoveL5 = depth - 3
    if (gr(correctMoveL5) == gr(wrongMoveL5))
  wrongDecisionsL5+=
     numOfPosAtDepth[depth] * (1 - averageCorrect[depth])
end
pathology = wrongDecisionsL5 / wrongDecisionsL1
gain = (1 - wrongDecisionsL5) / (1 - wrongDecisionsL1)
```

The granularity (*gr*) was applied in the following way: $h(p) = gr(h^*(p))$ and $h(n)=gr(h^*(n))$; see Figure 5. The granularity function *gr(number)* in the code transforms *number* according to the chosen granularity, as described in Section 5.1.

*numOfPosAtDepth[depth]* represents the number of all the positions *depth* moves away from the goal, as presented in Figure 2. The obtained results using the code for each particular granularity are presented in Table 1.

*averageCorrect[depth]* represents the probability of choosing the correct move when deciding randomly i.e., when the heuristic values with reduced granularity are equal for the wrong and the correct moves/positions; for example, if there are two positions and one is leading to the best solution and the other not, this factor is 1/2. If two of three positions are leading to the optimal solution, the factor is 2/3.

| $g$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $pat_{5,1}$ | 0.84 | 1.18 | 0.87 | 1.02 | 1.04 | 1.05 | 0.98 | 0.97 | 1.12 | 0.98 | 1.05 | 0.97 | 0.90 | 0.83 | 0.95 | 1.14 |
| $\text{gain}_{5,1}$ | 2.71 | 0.09 | 2.09 | 0.94 | 0.92 | 0.91 | 1.03 | 1.03 | 0.91 | 1.01 | 0.97 | 1.01 | 1.03 | 1.03 | 1.01 | 0.99 |

*Table 1: The degree of pathology and the search gain for the true values of positions*

At 7 out of 16 granularities in Table 1, the search is pathological even though there is no introduced noise in the heuristic function at all; just the true value is transformed according to the chosen granularity. If the values of the positions were corrupted by neither noise nor low granularity, the error when searching to all depths would be 0, $pat_{5,1}$ would be undefined and $gain_{5,1}$ would be 1. A deep search is intended to combat noise, and when noise is the main reason for error, it does this successfully, as evidenced by the lack of pathology at high granularities. However, when the main reason for error is a low granularity, a deep search does not reduce it – Table 1 indicates that sometimes a deeper and sometimes a shallower search performs better. In our experiments at small granularity, this effect overwhelms the noise-reducing effect of a deep search, even as reasonable noise is introduced, causing fluctuations at low granularities.

Although the fluctuations might seem random, Figure 12 indicates that at granularity 2 the ratio between the number of pathological positions (with 16 and 17 moves to the goal) and non-pathological positions (with 20 and 21 moves to the goal) determines the final result, since other positions make the search random. Therefore, the non-even distribution of the positions at different number of moves to the goal corresponds to those fluctuations.

We can also observe in Table 1 that as the granularity increases, the gain approaches 1. This happens because the higher the granularity, the fewer errors it causes at both search depths, bringing $gain_{5,1}$ ever closer to 1.

One might argue that granularity could be applied in a different way, e.g., by calculating $gr(4) + gr(h^*(p))$, but calculations reveal that the fluctuations persist. Furthermore, the results are practically the same as those in Table 1 if instead of the program code in this section the generic heuristic function with Gaussian noise close to zero is used.

**Observation 6**: If no noise is introduced in the generic heuristic function ($\sigma = 0$), the search gain and the degree of pathology fluctuate for granularities up to 17 for the usual 8-puzzle. For granularities greater than 17, the gain is 1 and the degree of pathology is undefined.

**Explanation 1:** Section 5.3 indicates that the fluctuations observed in Figures 6 to 11 with granularity at values up to 10 are caused by the introduced granularity and the uneven distribution of true values of positions.

It seems that the difference between fluctuations up to granularities 10 in Explanation 1, and 17 in Observation 6 stems from the added noise in the generic function.

## 5.4. Tentative explanation as to why higher granularity decreases pathology

The degree of pathology for a given search problem and a given heuristic function can be analytically calculated using the following procedure. For the start, the analysis will be devoted to a node $v$ in which one can make exactly one correct and one wrong move that lead to the nodes $v_c$ and $v_w$ respectively. First, suppose we know (or can approximate accurately enough) the probability distribution of the heuristic values for each group of nodes with the same true value. We will denote the probability distribution of the heuristic value in node $v_c$ with $p_c$ and the probability distribution of the heuristic value in node $v_w$ with $p_w$. The probability distributions of the backed-up values can be computed using the probability distributions of the heuristic values of the leaves and the structure of the search tree, assuming that the probability distributions are independent of each other. We will denote the probability distributions of backed-up values from level *j* in nodes $v_c$ and $v_w$ with $p_{c'}$ and $p_{w'}$ respectively. Using these assumptions the pathology is calculated using the following formula:

$$pat_{1,j} = \frac{1 - \sum_{k=-\infty}^{\infty} p_c(k)\left(0.5 p_w(k) + \sum_{i>k+1} p_w(i)\right)}{1 - \sum_{k=-\infty}^{\infty} p_{c'}(k)\left(0.5 p_{w'}(k) + \sum_{i>k+1} p_{w'}(i)\right)} \quad (8)$$

The summation in the numerator gives us the probability of choosing the correct move using the values of the heuristic function in nodes $v_c$ and $v_w$. Note that the probability is 1 if the heuristic value in $v_c$ is smaller than the heuristic value in $v_w$, 1/2 if the values are the same, and 0 otherwise. The summation in the denominator gives us the probability of choosing the correct move using the backed-up values.

If there are nodes in the search graph that are of different type (e.g., 2 correct and 1 wrong move), the equation must be corrected accordingly. The degree of pathology can be calculated as a weighted sum of the pathologies over all the types of nodes. The final result using the probability distributions from the 8-puzzle is the same as presented in Figure 7.

The probability density of the heuristics of the correct and wrong moves can be computed analytically or numerically. The probability density for the usual 8-puzzle where one can make one correct and one wrong move is presented in Figure 13 on the vertical axis. The difference between the true value of the root of the search tree and the backed-up value is presented on the horizontal axis. Similar analyses of other combinations and not just one correct and one wrong move at depth 1 enable the following simplified reasoning as to why smaller granularity hurts the estimation at depth 5 more than at depth 1:

**Explanation 2:** Since the observed and the computed probability densities are narrower (i.e., have smaller standard deviation) at depth 5 than at depth 1 for the usual 8-puzzle (see Figure 13), the introduced smaller granularity, i.e., bigger buckets of default estimation, causes a relatively greater decrease of the quality of estimation at depth 5 than at depth 1.

To intuitively understand Explanation 2, consider an extreme case where the two distributions at level 5 are close to the Dirac delta function: a sufficient granularity will distinguish perfectly, but a bucket including the two delta functions would render the decision random.

In another example, the computations for the case with one wrong and one correct move reveal that the ratio between probability of a correct move at $g = 2$ and $g = \infty$ is 0.71 at level 5 and 0.78 at level 1 again indicating that the small granularity hurts the deeper search more than it hurts the shallower search. For the other 5 types of nodes and the weighted average over all the types, a similar relation was found.

The probability densities are stiffer at greater depths because of the minmin algorithm, which calculates the backed-up values using the minimum function. A similar but analytical explanation for the minimax algorithm was presented in [Luštrek 2007].

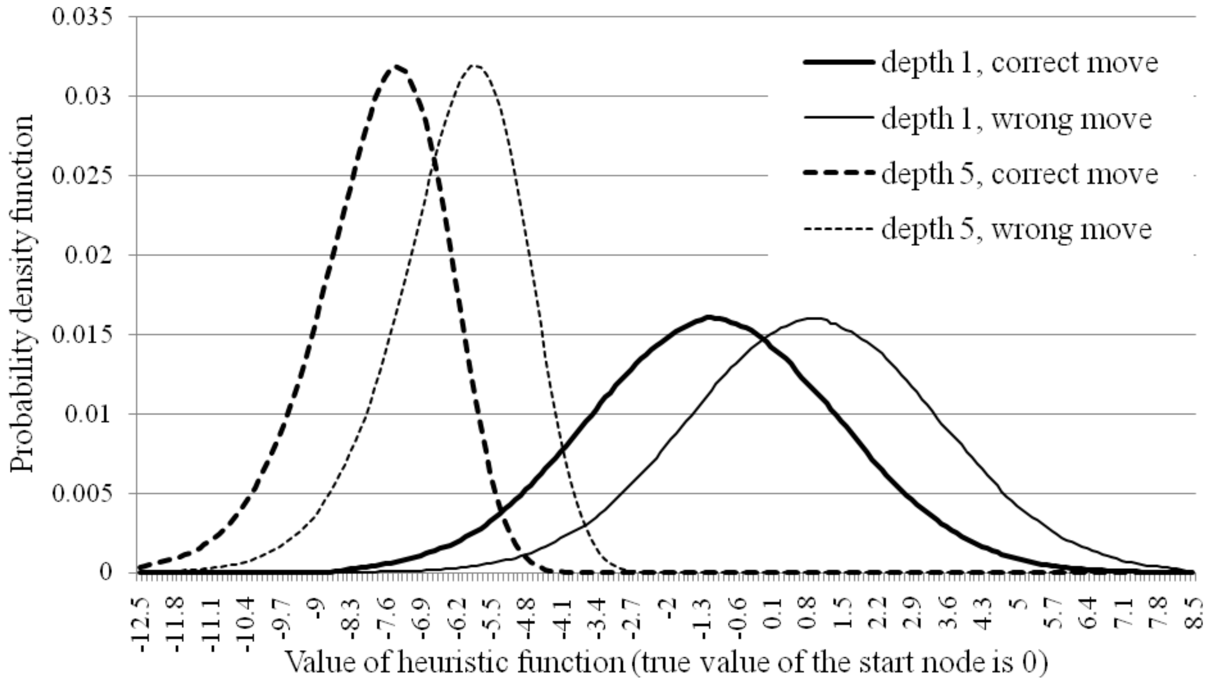



*Figure 13. The probability density functions of the heuristic function at level 1 and 5 for the usual 8-puzzle searches where the first move leads to one correct and one wrong move*

### 5.5 The amount of heuristic error

To analyze the influence of the amount of heuristic error on the pathology for the usual 8-puzzle at different granularities, we varied the error of the generic heuristic function. The error used in this paper is modeled by the Gaussian noise with $\sigma = 2.5$. In Figures 14 and 15, this noise is compared to twice smaller ($\sigma = 0.125$) and twice larger ($\sigma = 5$) noise and very small ($\sigma = 0.1$ ) and very large ($\sigma = 20$) noise at granularities from 2 to 2048.

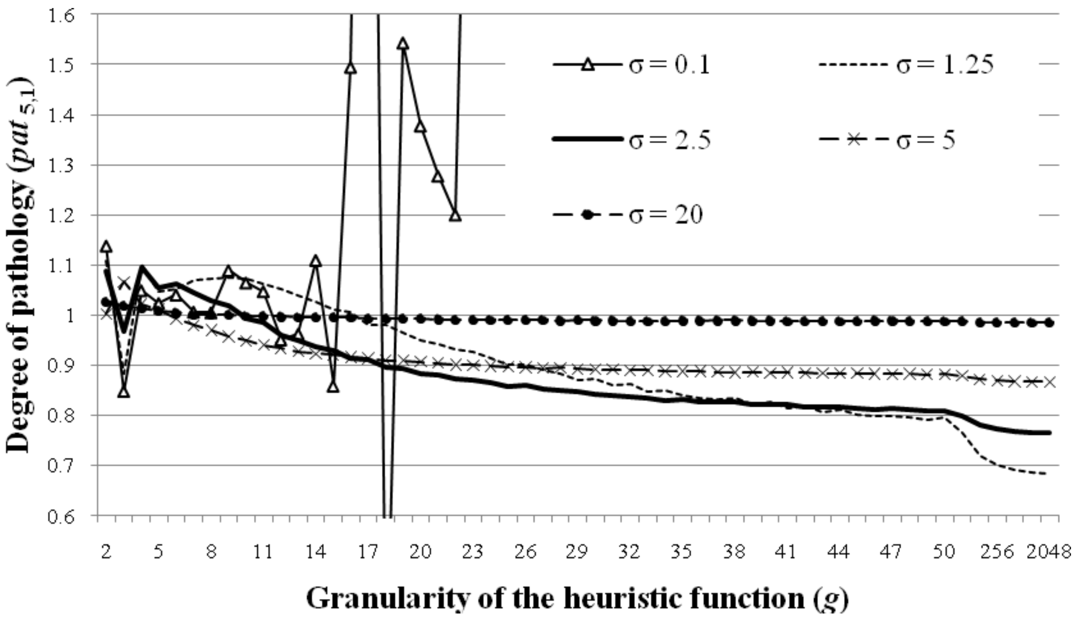



*Figure 14. The influence of the granularity on the degree of pathology for various amounts of heuristic error of the generic heuristic function in the usual 8-puzzle*

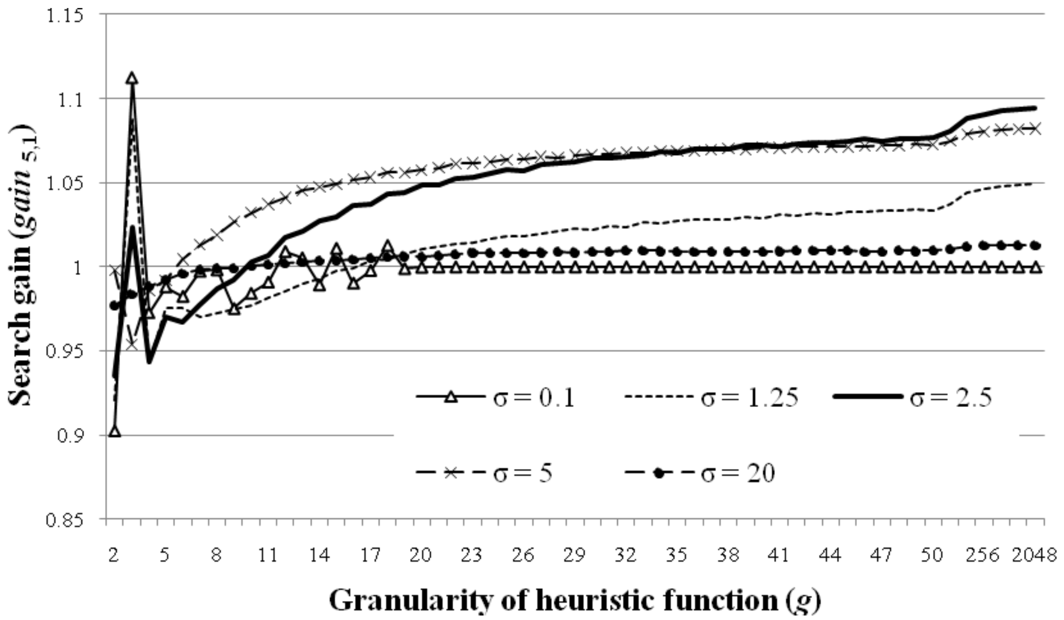



*Figure 15. The influence of the granularity on the gain for various amounts of heuristic error of the generic heuristic function in the usual 8-puzzle*

For $\sigma$ = 20, the degree of pathology and the gain are close to 1 (see Figure 14), since the differences between the positions' heuristic values are diminished by the noise, thus making the search essentially random. For $\sigma$ = 0, the gain and the degree of pathology were identical to Table 1 (not presented in Figures 14 and 15). For $\sigma$ = 0.1, the gain at 18 is not 1 anymore, as in the case of no noise, and the pathology remains computable up to 24. The degree of pathology and the gain for $\sigma$ = 0.1 fluctuate in a similar way as in Table 1 until $g$ = 17, since there are very few noise-introduced errors. For $g \geq 17$ the fluctuations of the degree of pathology increase wildly because there are no granularity-introduced errors and only a few noise-introduced errors (for example, if there are two errors at depth 5 and one error at depth 1, the degree of pathology is 2) and after $g$ = 24 there are no errors of any kind rendering the degree of pathology uncomputable. At moderate, i.e., reasonable noise ($\sigma$ = 1.25, 2.5 and 5), the search gain and the degree of pathology behave similarly – they fluctuate at a smaller granularity and tend to improve with larger granularity.

Observing the increase of gain in Figure 15, at small noise the errors are few, so there is little room for improvement. Large noise masks the differences in the position values so effective that a deeper search cannot uncover them.

**Observation 7**: Additional measurements (e.g., with $\sigma$ = 0, 0.01, 0.02, 0.05, 0.1, 0.5, 1, 1.25, 2, 2.5, 3, 4, 5, 10, 15, 20; varying also $i$ and $j$) show that the observed gain and the degree of pathology behave similarly as in Figures 14 and 15: large noise makes the search random, small noise makes the search similar to Table 1, with the exception that the degree of pathology fluctuates wildly until it becomes incomputable. Also observed, the gain of the deeper search for the usual 8-puzzle is the largest (and the degree of pathology, smallest) at moderate noise.

### 5.6 Optimistic and pessimistic heuristic functions

In this section, five heuristic functions are considered, their influence on pathology and the relation to the granularity. One of them is the generic function, two of the remaining four are optimistic and two are pessimistic. Optimistic heuristic functions always underestimate the true value and are often used because complete heuristic search algorithms such as A* need them to provide optimal solutions. However, past analyses showed that pessimistic heuristic functions outperform optimistic ones in an incomplete search and that they are also less prone to the pathology (Sadikov and Bratko 2006). Pessimistic, and especially optimistic heuristic functions, are usually easier to design than balanced ones, because it is often easy to compute some upper or lower bounds for the quality of nodes in the search tree.

Generic optimistic and pessimistic heuristic functions were obtained by subtracting or adding (respectively) the absolute value of the Gaussian noise $|e|$ from/to the true values $h^*(n)$. The pathology of the balanced (middle in Figure 16 a), optimistic (left) and pessimistic (right) generic heuristic functions was compared to the pathology of two real-life heuristic functions: the Manhattan-distance and Sadikov's heuristic function (Sadikov and Bratko 2007). Figure 16a shows the probability density functions for the balanced, optimistic and pessimistic generic heuristic functions. Figure 16b shows the percentages of the positions with a given heuristic error for the Manhattan-distance and Sadikov's heuristic functions. One can see that the former is indeed optimistic (the errors are all negative, which means that the heuristic values are smaller than the corresponding true values) and the latter pessimistic (the errors are positive).

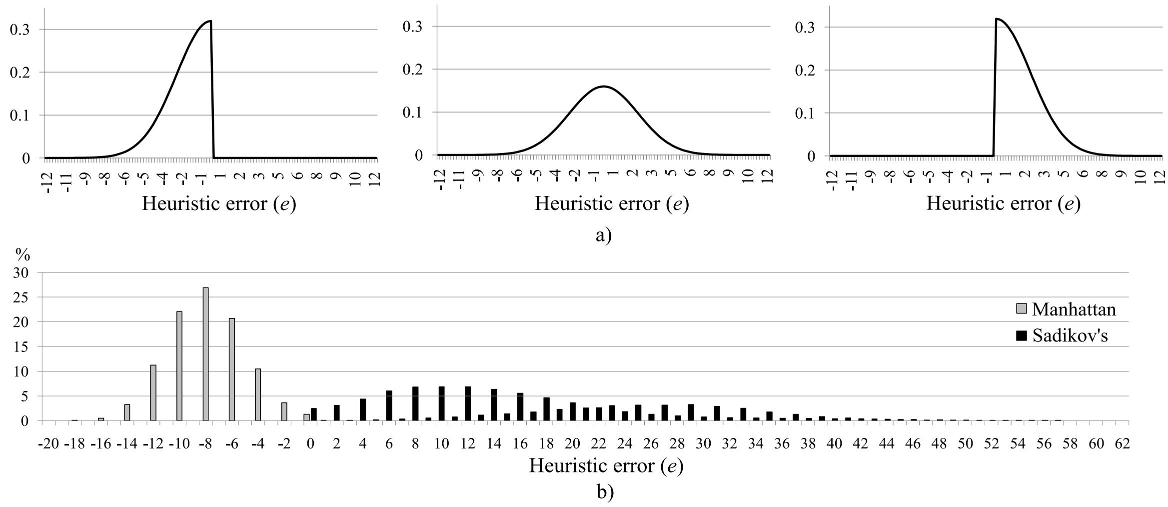



*Figure 16. The distributions of the heuristic values of the five heuristic functions compared in this subsection*

The Manhattan-distance heuristic function is an optimistic heuristic function that computes the minimum number of moves that would be required to solve the puzzle from a given position if the tiles could move one over the other. In other words, it is the sum of the Manhattan distances of the tiles' locations in a given position to their locations in the solved puzzle. The Manhattan-distance heuristic function generates integer values between 0 and 21.

The idea of Sadikov's pessimistic heuristic function for the $n^2$-puzzle is that the problem can be recursively decomposed into two easier sub-problems. The first problem is to solve the first line and the first column of a given puzzle without paying attention to the rest of the puzzle. The second problem is to recursively solve the rest of the puzzle, which is equivalent to solving the $(n\text{-}1)^2$-puzzle. Sadikov's pessimistic heuristic function returns the number of moves needed to solve a given 8-puzzle with the recursive procedure, which is larger or equal to the number of moves for an optimal solution.

The granularity of the five heuristic functions described in this subsection was reduced, as described in Section 5.1. The degree of pathology with respect to the granularity in the usual 8-puzzle is shown in Figure 17. The studies of gain revealed reverse relations.

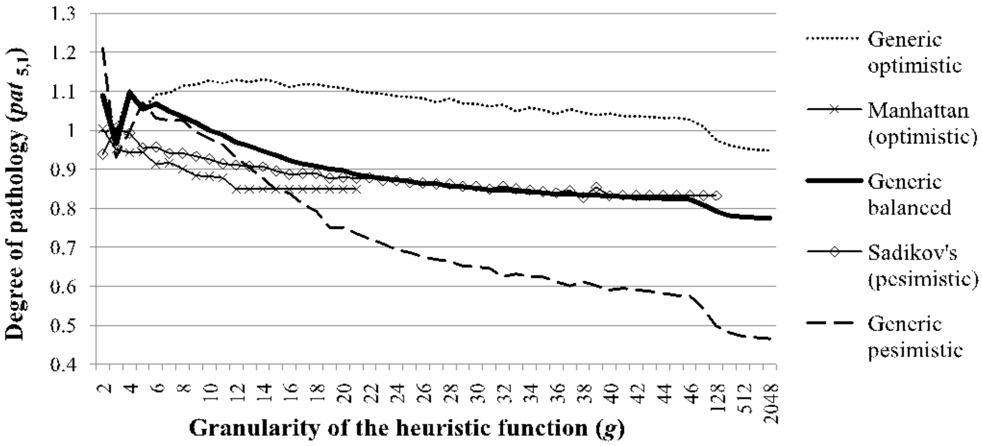



*Figure 17. The influence of the granularity of the heuristic function on the degree of pathology for optimistic, pessimistic and balanced heuristic functions in the usual 8-puzzle*

**Observation 8:** The observation that a decreased granularity in the 8-puzzle increases the degree of pathology and decreases the gain is valid for all five heuristic functions studied in this section: for the generic one and its pessimistic and optimistic variants as well as for the two real-life heuristic functions – Manhattan-distance and Sadikov's.

The curves in Figure 17 also confirm that pessimistic heuristic functions are less prone to the pathology than optimistic ones. The degree of pathology of the balanced generic heuristic function is higher than that of the pessimistic generic heuristic function, but smaller than that of the optimistic generic heuristic function. Although the optimistic Manhattan-distance heuristic function is seemingly less pathological than the Sadikov's pessimistic function, this is only because of the latter's larger maximum granularity – when comparing the degree of pathology at the maximum granularity, Sadikov's function is less pathological. But, since greater granularity in itself can influence the degree of pathology, the conclusion about the influence of the pessimistic function in this case is less undisputable, as is the case when comparing the versions of the generic function.

That pessimistic heuristic functions are more successful than the optimistic ones was already reported in previous studies (Sadikov and Bratko 2006), which also provided an explanation (Luštrek and Bulitko 2008); however, this was only done for artificial heuristic functions. In addition, in (Sadikov and Bratko 2006), the central theme of the paper was the success of optimistic versus pessimistic heuristic functions. In this section we also confirm their findings, but the emphasis is on the relation to the granularity, which turned out to be universal – a small granularity increases pathology.

## 6. The influence of the search tree on the pathology

The properties of the search tree are a part of the problem to be solved by heuristic search, and as such are not under the control of the one solving it. Because of that, their influence on the pathology is less important from the practical perspective than the influence of the heuristic function. Such properties are, however, important factors affecting the minimax pathology, so we studied them in the 8-puzzle as well.

### 6.1. The similarity of sibling nodes

The factor that is probably best known to affect the pathology in the minimax search is the similarity or (in)dependence of sibling nodes (Beal 1982; Bratko and Gams 1982; Nau 1983; Luštrek et al. 2006; Kaluža et al. 2007b).

**Definition 9:** A search tree has a high *similarity of sibling nodes* when the true values $h^*(n)$ of the descendants of a node *m* have similar values. This notion is quantified by the clustering factor (Sadikov 2005) given in Equation (9), which is roughly the ratio between the standard deviation of the true values of the sibling nodes averaged across all the possible parent nodes, and the standard deviation of the true values in the entire search tree.

$$Cf = \frac{\sqrt{\frac{1}{\sum_{i=1}^{N} b_i} \sum_{i=1}^{N} \sum_{j=1}^{b_i} \left(h^*(n_{ij}) - \frac{1}{b_i} \sum_{k=1}^{b_i} h^*(n_{ik})\right)^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(h^*(m_i) - \frac{1}{N} \sum_{j=1}^{N} h^*(m_j)\right)^2}} \tag{9}$$

In Equation (9) mi stands for a node in a search tree, nij stands for the j-th descendant of the node mi, bi for the branching factor of the node mi, and N for the number of nodes in the search tree.

The clustering factor is small when the sibling nodes are similar, because in that case the standard deviation of the sibling nodes' values is much smaller than the standard deviation of all the values.

In order to observe the effect of the similarity of the sibling nodes on the pathology, the degree of pathology for all the variants of the 8-puzzle was compared. Figure 18 shows the degree of pathology with respect to the clustering factor using the generic heuristic function for three different granularities. A linear approximation for each granularity is also plotted.

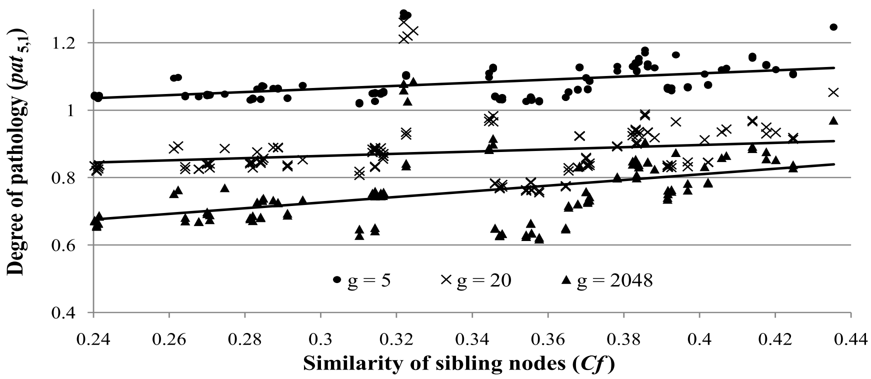



*Figure 18. The influence of the similarity of sibling nodes on the degree of pathology using the generic heuristic function with three different granularities*

In Figure 18 one can see that the degree of pathology decreases with the similarity of the sibling nodes (note that a smaller clustering factor means a larger similarity of the sibling nodes). As before, a larger granularity decreases the degree of pathology. The observed similarity-to-pathology relation is not very strong, for which there are at least two reasons. Firstly, it is not possible to observe the influence of the similarity of the sibling nodes alone, because the variants of the 8-puzzle differ in other factors, too. And secondly, the similarity in the figure ranges from 0.44 to 0.24, whereas in theoretical models it ranges approximately from 0.2 to 1.0 (Kaluža et al. 2007b). The explanation for the relation between the pathology and the similarity of the sibling nodes given for synthetic search trees (Luštrek 2007) probably holds for the 8-puzzle variants as well.

Graphs that included only the data for the variants of the 8-puzzle with an equal average branching factor were also plotted, again showing the decreasing trend of the degree of pathology with respect to the similarity of the sibling nodes. Another experiment was performed using the Manhattan-distance heuristic function (described in Section 5.4) adjusted to the directions of the move used by the usual and non-usual variants of the 8-puzzle. The results were again similar to those in Figure 18. Studies of the gain revealed the reverse relation to the pathology.

**Observation 9:** Analyses of the variants of the 8-puzzle indicate that a greater similarity, in general, decreases the degree of pathology and increases the gain, although the relation is not very strong.

### 6.2. The branching factor

The branching factor of the search tree is another factor known to affect the pathology (Nau 1979; Luštrek et al. 2006; Kaluža et al. 2007a, 2007b; Luštrek 2007). The branching factor of each of the variants of the 8-puzzle was averaged over all the solvable positions. The average branching factors for the 8-puzzle variants range from 1.56 to 4.44. There are 13 groups of 8-puzzle variants with the same branching factor, each consisting of 1 to 21 variants. Figure 19 shows the average degree of pathology of a group with respect to the branching factor using the generic heuristic function with three different granularities.

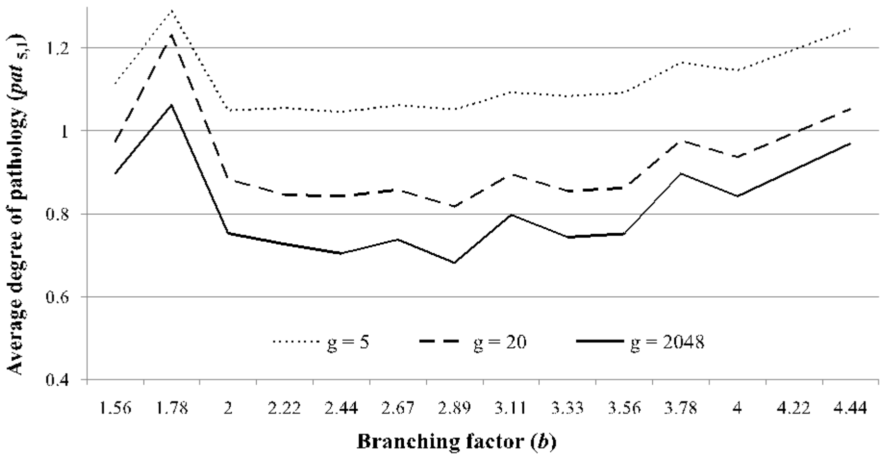



*Figure 19. The influence of the branching factor on the degree of pathology using the generic heuristic function with three different granularities*

No real conclusion about the influence of the branching factor on the pathology can be made from Figure 19. A graph similar to the one in Figure 19 was plotted for the adjusted Manhattan-distance heuristic function, but again no clear conclusion could be drawn. If the two groups of the 8-puzzle variants with the lowest branching factors (1.56 or 1.78) are removed from the graph, the average degree of pathology of the remaining 11 groups is increasing with the increasing branching factor, as expected and explained before (Luštrek 2007). The problematic two groups are the ones with just three move directions allowed. They also have an additional peculiarity: when solving them, every four moves there is only one legal move, whereas the remaining groups always have at least two possible moves. Analyses of the gain instead of the degree of pathology revealed basically inverse relations.

## 7. The percentage of pathological positions in the 8-puzzle

In order to find out how often the pathology occurs in the 8-puzzle, the number of pathological positions was computed. The probability of a wrong move at search depths 1 and 5 using the generic heuristic function with various granularities was computed for each solvable start position of the usual 8-puzzle. The results are presented in Figure 20. Note that the percentage of pathological positions does not directly correspond to the degree of pathology in Figure 7 where the probability of a wrong move is summed. The values on the x-axis denote the granularity of the generic heuristic function and the height of the bars denotes the percentage of positions belonging to one of three groups. The first group consists of the positions in which a depth-1 and depth-5 search result in the same probability of selecting an optimal move. The second group consists of pathological positions in which a depth-1 search results in a higher probability of selecting an optimal move. The third group consists of positions in which searching to depth 5 is worthwhile, since it increases the probability of selecting an optimal move.

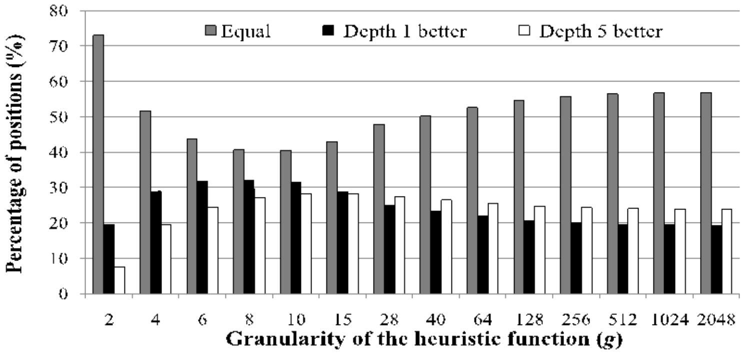



*Figure 20. Percentage of pathological and non-pathological positions at a given granularity of the generic heuristic function in the usual 8-puzzle*

Figure 20 shows that for granularities over 15, the percentage of pathological positions is smaller than the percentage of positions where a deeper search is beneficial. For granularities between 2 and 15 there are more pathological than non-pathological positions. The number of positions where both depths give equally good results increases with $g$ larger than 9.

The experiments presented in Figure 20 were repeated using the Manhattan-distance and Sadikov's heuristic functions with unrestricted granularity. With the Manhattan-distance heuristic function there were 49.3 % of positions where the gains of both depths were equal, 19.7 % of pathological positions and 31.0 % of positions where a depth-5 search produced a higher gain. With Sadikov's heuristic function, there were 67.0 % of positions where the gains of both depths were equal, 13.4 % of pathological positions and 19.5 % of positions where a depth-5 search produced a higher gain. For comparison, the three percentages for the generic heuristic function are 56.8 %, 19.3 % and 23.9 %. The percentage of positions where a deeper search is preferable is the largest with the Manhattan-distance heuristic function (both in absolute terms and compared to the percentage of pathological positions). The absolute percentage of such positions is the smallest for Sadikov's heuristic function, which is the best of the three for solving large $n^2$-puzzles (Sadikov and Bratko 2007), and this percentage compared to the percentage of pathological positions is the smallest for the generic heuristic function.

**Observation 11.** In the usual 8-puzzle at unrestricted granularity, the following ratios between positions with better gains at search 5 than at search 1 are obtained: the Manhattan-distance heuristic function 31.0% – 18.7%, Sadikov's function 19.5% – 13.4%, and the generic function 23.9% – 19.3%.

## 8. Conclusion

We have studied the 8-puzzle domain to find out under which conditions the search gain and the degree of pathology of a deeper heuristic search increase and when the search pathology occurs. To investigate the effect of the similarity of sibling nodes and the branching factor on the search gain, the original 8-puzzle was extended by adding diagonal moves and then allowing a specific subset of potential moves. The obtained 126 variants of the 8-puzzle differ in terms of the similarity and the branching factor. In most of the cases we compare the search depths 1 and 5, i.e., what is the success rate of two programs, one searching to depth 1 and the other to depth 5? In the majority of the tests, all the positions are taken into account, even though some of them lead to the goal in less than 5 moves. In addition, several tests were performed when searching to depths $j$ and $i$, indicating that conclusions for the analyses at depth 1 and 5 are valid for several other search depths.

The conclusions are as follows:

- Increased granularity of the heuristic function increases the search gain and decreases the degree of pathology. This was previously observed in the minimax search, but not yet in the single-agent search.
- The most important single factor to improve search gain and avoid pathology is a high percentage of goal positions and positions with low or no heuristic error (compared to the reasonable heuristic error at smaller depth) in the search space. Other factors, such as branching or similarity, did not turn out to be essential in the variants of the 8-puzzle. This observation was initially proposed nearly 30 years ago (Bratko and Gams 1982; Pearl 1983) and later regarded as less important.
- Very large or very small heuristic error both decrease the search gain; therefore, at some reasonable noise a deeper search is most advantageous. This was observed in the minimax search under specific conditions, but not in the single-agent search until now.
- Pessimistic heuristic functions have higher search gains than optimistic ones. This was known for artificial heuristic functions in the 8-puzzle, but in this paper we also analyze real-life heuristic functions.
- An increased similarity of the sibling nodes in the search tree increases the search gain. This was again observed in the minimax search, but not in the single-agent search. The relation is not very strong, probably because of the small variation in the similarity in our experiments, and also because it is not possible to vary the similarity without changing other properties of the 8-puzzle variant as well.
- A decreased branching factor of the search tree slightly increases the search gain, which is also in agreement with the minimax search, but the findings are inconclusive.

Both the minimin and minimax search are similar with respect to the influence of various factors on the search gain and the pathology, and can be studied within a common framework. Although several observations in this paper were previously reported in the minimax search, some of them are altogether new, e.g. the overwhelming effect of early terminations and perfectly-evaluated positions and the resulting pathology without them.

The pathology is usually considered a rare occurrence. However, when measured per position, this is not necessarily the case, as the 8-puzzle demonstrates. In our experiments, the percentage of pathological positions for three different heuristic functions was at least half of the percentage of positions where it is worthwhile to search deeper, and often the percentages were very close. For example, the fraction of all positions in the 8-puzzle where searching to depth 5 is better than to depth 1, compared to those where the reverse is true, 31.0% versus 19.7% if Manhattan-distance heuristic function is used. In addition, the percentage of positions where both depths were equally good was quite large Therefore, the advantages of a deeper search are not as clear as it is often thought and the pathology can have a substantial impact on the performance of incomplete search algorithms.

Even more important than the pathology itself, the analysis of the influence of the various factors on the search gain in this paper might provide some general clues for designing better heuristic functions and search algorithms:

- When designing heuristic functions, one should strive for a large granularity and should not ignore nuances when evaluating positions.
- A deeper search can be trusted more if there are many positions with clear differences in the heuristic values in the search space, i.e. perfectly or near-perfectly estimated positions.
- When the heuristic error is very large or very small, a deeper search might not be worthwhile – if the computation time is at premium, it might be better to spend it more on evaluating positions than on searching deeper.
- For an incomplete single-agent search using minimin algorithm, pessimistic heuristic functions are more suitable than the optimistic or the balanced ones.
- When a small share of positions with trustworthy estimations appear in the search space, and in addition a small granularity of the heuristic estimation is chosen, a small similarity of the sibling nodes and a large branching factor turns out, one should not automatically assume that a deeper search is beneficial.

Probably the most interesting direction for further research is attempting to adjust the depth of the search dynamically per position. By not searching deeper than necessary, one would save CPU time to use in cases when a deeper search is worthwhile or on a more sophisticated position evaluation. In addition, by not searching too deep in pathological positions, one would obtain better solutions. We used machine learning to determine in which positions a deeper search is necessary, but our attempt was not met with much success. The most successful classifiers were those using attributes that can only be computed if all the positions in the puzzle are solved first, which is obviously impractical. When practical attributes were used, it was possible to identify only some small groups of positions that were mostly pathological; therefore, the practical improvement using this information was rather small. However, the efforts to adjust the depth of the search dynamically were more successful in the domain of path-finding (Bulitko, Luštrek, Schaeffer, Björnsson and Sigmundarson 2008), so this direction of research deserves more attention.

## Acknowledgement

This work was supported by the Slovenian Research Agency under the Research Programme P2-0209 Artificial Intelligence and Intelligent Systems. We thank Ivan Bratko, Boštjan Kaluža and Aleš Tavčar for valuable comments and contributions, Dana Nau for related work, and the anonymous referee for the valuable comments that enabled us to make significant improvements to the paper.

## References

Beal, D.F. (1980), 'An analysis of minimax', in Proceedings of the Advances in Computer Chess 2, Edinburgh, UK, pp. 103–109.

Beal, D.F. (1982), 'Benefits of minimax search', in *Proceedings of the Advances in Computer Chess 3,* Oxford, UK, pp. 17–24.

Box, G.E.P. and Muller, M.E. (1958), 'A note on the generation of random normal deviates', *The Annals of Mathematical Statistics, vol. 2*, 29, 610–611.

Bratko, I. and Gams, M. (1982), 'Error analysis of the minimax principle', in *Advances in Computer Chess 3,* Oxford, UK, vol 3, pp. 1–15.

Bulitko, V. (2003), 'Lookahead pathologies and meta-level control in real-time heuristic search', in *Proceedings of the 15th Euromicro Conference on Real-Time Systems*, Porto, Portugal, pp. 13–16.

Bulitko, V., Li, L., Greiner, R. and Levner, I. (2003), 'Lookahead pathologies for single agent search', in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, pp. 1531–1533.

Bulitko, V., Luštrek, M., Schaeffer, J., Björnsson, Y., and Sigmundarson, S. (2008), 'Dynamic control in real-time heuristic search', *Journal of Artificial Intelligence Research*, 32, 419–452.

Heinz, E. A. (2001), 'New self-play results in computer chess', in *Lecture Notes In Computer Science, vol 2063*, pp. 262–276.

Hernández, C, and Meseguer, P. (2008), 'Combining lookahead and propagation in real-time heuristic search', in *Proceedings of The First International Symposium on Search Techniques in Artificial Intelligence and Robotics*, Chicago, USA.

Junghanns, A., Schaeffer, J., Brockington, M., Bjornsson, Y., and Marsland, T. (1997), 'Diminishing Returns for Additional Search in Chess', in *Proceedings of the Advances in Computer Chess 8*, Maastricht, The Netherlands, pp. 53–67.

Kaluža, B., Luštrek, M. and Gams, M. (2007a), 'Minimax pathology in synthetic trees and Pearl game'', in *Proceedings of the 10th international multiconference Information Society*, Ljubljana, Slovenia, pp. 84–87.

Kaluža, B., Luštrek, M., Gams M. and Tavčar, A. (2007b), 'Pathology in Minimax Searching', in *Proceedings of the 16th International Electrotechnical and Computer Science Conference*, Portorož, Slovenia, pp. 111–114.

Korf, R.E. (1990), 'Real-time heuristic search', *Artificial Intelligence*, vol. 2–3, 42, 189–211.

Kunkle, D.R. (2001), 'Solving the 8 Puzzle in a Minimum Number of Moves: An Application of the A* Algorithm', *Introduction to Artificial Intelligence*, College of Computing and Information Sciences, Rochester Institute of Technology.

Luštrek, M. (2005), 'Pathology in single-agent search', in *Proceedings of the 8th international multiconference Information Society*, Ljubljana, Slovenia, pp. 345–348.

Luštrek, M. (2007), 'Pathology in heuristic search algorithms, ', Ph.D. thesis, University of Ljubljana, Faculty of computer and information science.

Luštrek, M. and Bulitko, V. (2006), 'Lookahead Pathology in Real-Time Path-Finding', in *Proceedings of the National Conference on Artificial Intelligence, Learning For Search Workshop*, Boston, USA, pp. 108–114.

Luštrek, M., Gams M. and Bratko, I. (2006), 'Is Real-Valued Minimax Pathological?', *Artificial Intelligence*, vol 6–7, 170, 620–642.

Luštrek, M. and Bulitko, V. (2008), 'Thinking Too Much: Pathology in Pathfinding', in *Proceedings of the 18th European Conference on Artificial Intelligence*, Patras, Greece, pp. 899–900.

Nau, D.S. (1979), 'Quality of decision versus depth of search on game trees', Ph.D. thesis, Duke University.

Nau, D.S., (1983), 'Pathology on game trees revisited, and an alternative to minimaxing', *Artificial Intelligence, vol 1–2*, 21, 221–244.

Nau, D.S., Luštrek, M., Parker, A., Bratko, I. and Gams, M. (2010), 'When is it better not to look ahead? ', *Artificial Intelligence*, vol 174, 16/17, 1323-1338.

Neumann, J. von (1928), 'Zur Theorie der Gesellshaftsspiele', *Mathematische Annalen*, 100, 295–320.

Pearl, J. (1983), 'On the Nature of Pathology in Game Searching', *Artificial Intelligence, vol 20*, 4, 427–453.

Pearl, J. (1984), 'Heuristics: intelligent search strategies for computer problem solving', Boston MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Piltaver, R., Luštrek, M. and Gams, M. (2007), 'Search Pathology of 8-Puzzle', in *Proceedings of the 10th international multiconference Information Society*, Ljubljana, Slovenia, pp. 95–98.

Reinefeld, A. (1993), 'Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA*', in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, Chambéry, France, pp. 248–253.

Ratner, D. and Warmuth, M.K. (1986), 'Finding a shortest solution for the N × N extension of the 15-Puzzle is intractable', in *Proceedings of the AAAI National Conference on Artificial Intelligence*, Philadelphia, USA, pp. 168–172.

Ratner, D. and Warmuth, M.K. (1990), 'The ($n^2$ – 1)-Puzzle and related relocation problems', *Journal of Symbolic Computation*, vol 2, 10, 111–137.

Russel, J. and Norvig, P. (2010*), Artificial Intelligence, A Modern Approach*, Third Edition, Pearson Education.

Ryan, M., Solvability of the Tiles Game, http://www.cs.bham.ac.uk/~mdr/teaching/modules04/java2/TilesSolvability.html.

Scheucher, A. and Kaindl, H. (1998), 'Benefits of using multivalued functions for minimaxing', *Artificial Intelligence*, vol 2, 99, 187–208.

Sadikov, A. (2005), 'Error propagation in game search graphs', Ph.D. thesis, University of Ljubljana, Faculty of computer and information science.

Sadikov, A. and Bratko, I. (2006), 'Pessimistic Heuristics Beat Optimistic Ones in Real-Time Search', in *Proceedings of the European Conference on Artificial Intelligence*, Riva del Garda, Italy, pp. 148–152.

Sadikov, A. and Bratko, I. (2007), 'Solving 20x20 puzzles', in *Proceedings of Computer Games Workshop 2007*, Amsterdam, The Netherlands, pp. 157–164.

Thompson, K. (1986), 'Retrograde analysis of certain endgames', *ICCA Journal*, vol 3, 9, 131–139.

Wikipedia, Fifteen puzzle, http://en.wikipedia.org/wiki/Fifteen_puzzle.