The best work⋆

Arroni del Riego, Sergio[276341], Alex[1111−2222−3333−4444], and Manu[2222−−3333−4444−5555]

Oviedo University, Oviedo Asturias, Spain
https://uniovi.es

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: A* · PEA* · Search · heuristic · 8-puzzle.

1 Introduction

In this section, we will introduce the subject to be dealt with as well as a brief description of the rest of the sections of the work.

1.1 Description of the sections of the work

TODO

1.2 Description of the topic to be addressed

TODO

2 The 8-puzzle problem

TODO

3 The search algorithms

In this section, we will discuss about search algorithms, doing a distincion between uninformed and informed search algorithms. The uninformed algorithms that we will discuss are: Best First (BF). The informed algorithms that we will discuss are: A* and PEA*.

⋆ Degree in Software Engineering, University of Oviedo

3.1 Best first search (*BF*)

The uninformed search algorithms are those thay do not use any information about the problem, they just expand the next node in the frontier. The most important thing that differentiates the various uninformed algorithms is the method they use to expand the next node in the frontier. Hereafter, we will trate the Best First (BF) search algorithm.

We will start with the "Best First" (BF) search algorithm [1], which is the basis of informed search algorithms, such as A^*. BF is based on the idea of idea of expanding the node with the lowest cost. The cost of a node is calculated by the heuristic function $h(n)$ used.
In the BF algorithm, we have two list, the list "open" that this list contains the nodes that are not expanded yet, and the list "closed" that contains the nodes that are expanded:

1. Initialises the "open" list with the start node.
2. Initialises the "closed" list with the empty list.
3. While the "open" list is not empty:
   (a) Select the node with the lowest cost, following the given $h(n)$, in the "open" list and remove it from the "open" list.
   (b) If the node is the goal, stop.
   (c) If not, add the node to the "closed" list and expand its children.
   (d) For each child:
      i. If the child is in the "closed" list, do nothing.
      ii. If the child is not in the "open" list, add it.
      iii. If the child is in the "open" list, but the path to the child is better than the previous path, replace the child in the "open" list with the new child.
4. Return to Step 3.

You can see a more detailed description and a implementation of the algorithm in Russell's and Norvig's book [2].

3.2 A^*

The informed search algorithms are those that use the any information about the problem to expand the next node in the frontier.
The A^* algorithm is a variation of the BF algorithm, this was proposed by Peter E. Hart, Nils J. Nilsson and Bertram Raphael in their work in 1968 [3].
In A^*, use a function f to evaluate the nodes, this function in A^* is representate by $f^*(n)$ and this is the cost of the shortest path from initial to n conditional on passing through n. The function $f^*(n)$ (where n is any node) is defined as equation 1:

$$f^*(n) = g^*(n) + h^*(n) \tag{1}$$

Where $g^*(n)$ is the cost of the path from the initial node to the node n, and $h*(n)$ is the heuristic function that estimates the cost of the path from the node n to the goal node, you can see this function in more detail in Nilsson's book "Principles of artificial intelligence" [4]. In A* we talk about f*, g* and h*, but in most cases these are only estimates because it is very complicated for complex problems to know the exact values, if we knew them the algorithm would go straight to the goal. Instead we use the estimates f, g and h, so the function would be as we can see in equation 2:

$$f(n) = g(n) + h(n) \tag{2}$$

The A^* algorithm is as follows, we have two list, the list "open" that this list contains the nodes that are not expanded yet, and the list "closed" that contains the nodes that are expanded:

1. Initialises the "open" list with the start node.
2. Initialises the "closed" list with the empty list.
3. While the "open" list is not empty:
   (a) Select the node with the lowest cost, you can se the equation in 1, in the "open" list and remove it from the "open" list.
   (b) If the node is the goal, stop.
   (c) If not, add the node to the "closed" list and expand its children.
   (d) For each child:
      i. If the child is in the "closed" list, do nothing.
      ii. If the child is not in the "open" list, add it.
      iii. If the child is in the "open" list, but the $f(child)$ is better than the previous path, replace the child in the "open" list with the new child.
4. Return to Step 3.

One problem that have the A^* algorithm is that it can be very slow, because it can expand a lot of nodes, and this can be a problem if the problem has a lot of nodes. To solve this problem, we can use the PEA* algorithm.
If you want to see a more detailed description and a implementation of the algorithm, you can see Russell's and Norvig's book [2].

3.3 PEA^*

The PEA^* algorithm is a variation of the A^* algorithm, in fact it is faster than the base algorithm A^*. PEA^* is a not admitted algorithm, this means that it is not guaranteed to find the optimal solution, but it is very fast and it is very useful in practice. The PEA^* algorithm have a function, very similar to the A^* function, but it is not the same, this function is called $f^*_{PEA}(n)$ and is defined in equation 3:

$$f^*_{PEA}(n) = g^*(n) + h^*(n) * (1 + \epsilon) \tag{3}$$

Where $g^*(n)$ is the cost of the path from the initial node to the node n, $h*(n)$ is the heuristic function that estimates the cost of the path from the node n to the goal node and ϵ is a constant that is used to control the expansion of the nodes.
You can see this algorithm in more detail in Maria Rita's tesis [5].

4 Application of the A* and PEA^* algorithms to the 8-puzzle problem

TODO

5 Experimental study

TODO

6 Conclusions

TODO

7 DE AQUI PARA ABAJO = BASURA, LO DEJO DE EJEMPLO

Please note that the first paragraph of a section or subsection is not indented. The first paragraph that follows a table, figure, equation etc. does not need an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level) Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Fourth Level) The contribution should contain no more than four levels of headings. Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \tag{4}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

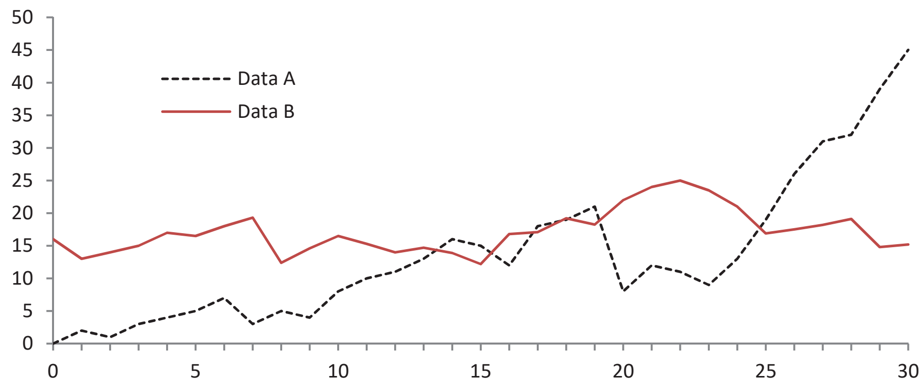

Fig. 1. A figure caption is always placed below the illustration. Please note that short captions are centered, while long ones are justified by the macro package automatically.

Theorem 1. *This is a sample theorem. The run-in heading is set in bold, while the following text appears in italics. Definitions, lemmas, propositions, and corollaries are styled the same way.*

Proof. Proofs, examples, and remarks have the initial word in italics, while the following text appears in normal font.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal [**?**]

References

1. J. Pearl, "Heuristics : intelligent search strategies for computer problem solving," p. 382, 1984.
2. P. R. Norvig and S. A. Intelligence, *A modern approach*, vol. 90. 2002.
3. P. E. Hart, N. J. Nilsson, and B. Raphael, *A formal basis for the heuristic determination of minimum cost paths*, vol. 4. IEEE, 1968.
4. N. J. Nilsson, *Principles of artificial intelligence*. Springer Science & Business Media, 1982.
5. M. R. Sierra Sanchez *et al.*, "Mejora de algoritmos de busqueda heuristica mediante poda por dominancia. aplicacion a problemas de scheduling," 2009.