A Heuristic Bioinspired for 8-Piece Puzzle

M O Machado[1, a], P A Fabres[2, b] and J C L Melo[3, c]
[1]UFMS/CPAN - Brazil, [2]UFMS/CPAN - Brazil and [3]UFRJ - Brazil

[a]murilo.machado@ufms.br; [b]phelipe.fabres@ufms.br; [c]cabrallima@ufrj.br

Abstract. This paper investigates a mathematical model inspired by nature, and presents a Meta-Heuristic that is efficient in improving the performance of an informed search, when using strategy A * using a General Search Tree as data structure. The work hypothesis suggests that the investigated meta-heuristic is optimal in nature and may be promising in minimizing the computational resources required by an objective-based agent in solving high computational complexity problems (n-part puzzle) as well as In the optimization of objective functions for local search agents. The objective of this work is to describe qualitatively the characteristics and properties of the mathematical model investigated, correlating the main concepts of the A * function with the significant variables of the metaheuristic used. The article shows that the amount of memory required to perform this search when using the metaheuristic is less than using the A * function to evaluate the nodes of a general search tree for the eight-piece puzzle. It is concluded that the meta-heuristic must be parameterized according to the chosen heuristic and the level of the tree that contains the possible solutions to the chosen problem.

1. Introduction

Works that uses the 8-Piece Puzzle Problem to introduce artificial intelligence are quite common. [1][2][3][4][5]. Mainly, the ones with focus in optimizing the amount of memory spend and the processing timing required by the computer to solving Stochastic problems, The search strategy A* is a model well discussed in the literature. The different properties and qualities resulting from the chosen heuristic show us that, if it is permissible and consistent, the search is optimal [6]. However, depending on the problem, memory usage and runtime may be equivalent to NP-Complete problems [7].

Some papers concentrate their efforts on the search for extensions of the two classic heuristics: h1 (quantity of pieces out of place) and h2 (distance of Manhattan). Be investigating the search strategies A * and AIA * (which is the iterative version of A *) and presenting a third heuristic with small modifications in the distance of Manhattan. Or, showing significant improvements in memory required [1] [8].

Proper heuristics offer optimum results for these problems, especially in the areas of Artificial Intelligence (IA) and Optimization of Algorithms [9]. Computational Intelligence (CI) has some interesting approaches to develop computational models, such as Neural Networks, Fuzzy Logic and Genetic Algorithms. Indeed, IC has been widely used in computer simulations and has shown to be successful in many models of real problems [10].

An important improvement in this context was the integration of CI with Bio-Inspired Computing. In Swarm Intelligence (SI), the main objective is to understand how the interactions between agents occur, and how these agents behave in the face of changes in their environment [11] [12]. The central hypothesis is that there is intelligence in the overall behaviour of a group of agents, although each member of this group follows natural rules without knowing who controls or guides the group [13] [14].

Ant colony models [15] and school of fish [16] has obtained important solutions to problems that require high computational power. Mathematical models are powerful tools for understanding and describing the actual physical phenomena. In order to validate mathematical models, computational models are widely used to simulate these phenomena. A robust computational model usually deals with a nonlinear problem with solutions that do not have trivial analytical results.

The Problem of the n-Piece Puzzle is an excellent example to investigate the performance of search algorithms, since it presents an exponential complexity in relation to the quantity of pieces of the puzzle. Different search strategies have been applied in solving this problem, where the amount of computational memory and processing time required to it are the main variables to be investigated.

The general objective of this article is to present new properties of a mathematical model [8], being described qualitatively these new characteristics and correlating with the concepts of the search strategy A *.

Also, it is intended to solve the problem of the Eight-Piece Puzzle using the mathematical model presented and comparing the results obtained with the search strategy A *. The amount of memory used will be the parameter of comparison between the models investigated.

2. The meta-heuristic

The search strategy A * [6] presents the function $f(n) = g(n) + h(n)$, where the cumulative cost of the search is provided by $g(n)$, and $h(n)$ is the heuristic that When the search is close to the solution.

When a search tree is used as the data structure, the number of expanded nodes and number of nodes stored in the border variable are the computational variables related to the complexity of the problem. The literature shows that the strategy A * is adequate for solving the problem of Puzzle n parts when chosen a good heuristic[17]. There are two permissible and consistent heuristics widely used in the investigation of this problem, with $h1$ and $h2$.

According to Russell [10], the simplest way to reduce memory requirements is to use AIA *. However the same problems (such as generation of repeated nodes) from iterative deepening are encountered. The mathematical model presented here was built upon the investigation of the behaviour of bird migration [18]

$$L(\alpha, V_a) = \frac{m * V_a^2}{2} - m * V_a * V_w * cos(\alpha) + \frac{k * V_a^2}{2} \tag{1}$$

The migration of birds has some interesting peculiarities, such as the "same destination" in numerous migratory journeys made by new-born birds, endowed with a very efficient orientation and until today not very understood. Attempts to explain phenomena of this nature are generally grounded in energy optimization theories, which induce (in certain configurations) that birds tend to minimize their energy expenditure.

From this model, a meta-heuristic was generated based on the Newton-Raphson method [10], and we used the operator $\partial L / \partial V_a / \partial^2 L / \partial V_a^2$ of the Hessian matrix to find the points of (1) and then an adjustment was made to the value f obtained by the strategy A *, resulting in equation 2.

$$f(n) = f(n) - \partial L / \partial V_a / \partial^2 L / \partial V_a^2 \tag{2}$$

This paper intends to investigate the performance of the metaheuristic (equation 2) in relation to the search strategy A * when the 8-Piece Puzzle is solved. Note that the basis for solving the problem is the Tree General Search Algorithm and what differs is the evaluation of tree nodes by metaheuristic or A * strategy.

3. Experiment

The experiment consists in solving the problem of the 8-piece puzzle by taking 1000 samples of initial states, which were obtained after 20 random movements from the problem solution (objective state) [19]. Thus, the existence of solutions among the 20 First Levels of the Search Tree is guaranteed.

From the meta-heuristic, two mathematical functions were generated to be applied in the experiment: L1, which is equation 2 with h1 inserted in the domain and L2, which is equation 2 with h2 inserted in the domain.

In the meta-heuristic, the domain of the generalized coordinates α and V_a will be parameterized according to the limits of the heuristic used, and the values of referring to variable g corresponding to the model A *, then the limits will be adjusted within the appropriate limit for the Problem investigated. In this way $h1$ can range from 0 to 8, and $h2$ can range from 0 to 26. The domain of the generalized coordinate α limited from 0 to 180 o will be divided by 8 linearly spaced points when used $h1$ and divided by 26 linearly spaced points when it traverses to $h2$. The generalized coordinate domain V_a of 0 to 40 m / s will be divided between 8 linearly spaced points when used $h1$ and divided by 26 linearly spaced points when $h2$ is used.

Samples were provided as input to the Tree General Search Algorithm using $h1$ and $h2$ (using the A * search strategy), and $L1$ e $L2$ (using metaheuristics). The tree is generated from the initial node, entered as input into the algorithm, and consists of the nodes results of the expansion and are stored on the edge, which is a sub tree with all possible states for the problem at that time.

As a result, we evaluated the evaluation of each node expanded and stored on the edge [10], then the average of the total amount of expanded nodes is calculated to solve the problem for each example.

In the evaluation of the obtained results, a table was constructed to present the average of the generated nodes when using each heuristic and distinct heuristic goal, and a figure to show the evaluation of the state spaces to each expanded node in the course of a search.

4. Results

Figure 1 shows the behaviour of the evaluation functions $f(n)$ (figure to the left) and L figure to the right) used in this article. The image of the evaluation function $f(n)$ is contained in a discrete and limited universe between 0 and 60. The image of the evaluation function L is contained in a continuous universe with the image limited between -10 and 40.

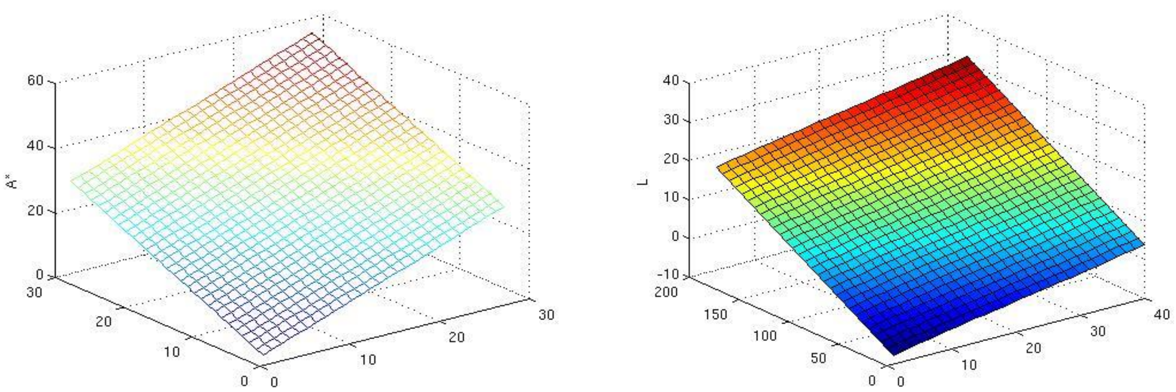

Figure 1. Figure 1 (a) shows the value of the function f (n) and Figure 1 (b) shows the value of the function L.

It is also observed that the voids contained in Figure 1 (a) are contemplated by Figure 1 (b) generating some zones of local maxima and minima, but without damaging the global growth presented by the function $f(n)$. It can also be observed that the evaluation of L does not have linear growth as the function $f(n)$. The grid on the evaluation surface of figure 1 (b) corresponds to the values obtained in equation 2 when the values of $g(n)$ and $h(n)$ are inserted. The values assigned in L to the constants m and k, and in how many parts and divided the domain of α and V_a are determinant to move this grid by the evaluation surface.

Figure 2 shows the evolution of the search space to each loop of the algorithm, showing the evaluation of each node that is inserted in the edge. The blue dots represent the upward growth of $f(n)$ and the green dots, referring to L, show some zones of minimum and maximum locations.

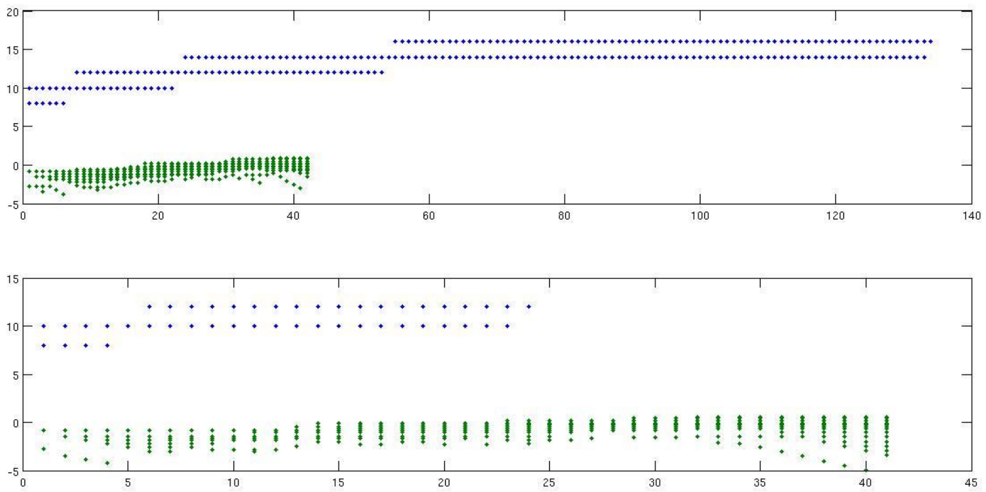

Figure 2. Figures 2 (a) and 2 (b) show the nodes contained in the space by the strategy A * and the nodes contained in the space with the Meta-Heuristic L, respectively.

The article focused on empirically investigating which better division of the domain of equation 2. Initially, the domain V_a was divided into 29 parts and the domain α into 26 parts and it was verified that the domain obtained good results when the value of the domain of V_a was decreased. Since h can not be greater than 26 and the sample used does not contain problems in depth greater than 20, there is a small probability of $h(n)$ being greater than 20. Following this hypothesis, the space of V_a was divided into smaller parts, and the results presented in table 1 (division into 18 parts) were verified.

It can be observed in figure 2 that the algorithm searches for the minimum of the function always choosing the node with the lowest evaluation of f or L. Since $f(n) > 0$ for all $g(n)$ and $h(n)$ the curve is ascending for $f(n)$ and descending to L since its nodes have negative evaluation in this domain. Figure 2 (a) shows 134 executions of $f(n)$ and 42 of L. Figure 2 (b) shows 24 executions of $f(n)$ and 41 of L Note that some nodes are overlapped by having same evaluation. The evaluation of the nodes by the function L is able to differentiate the nodes that had the same evaluation by the function $f(n)$. This can be observed in figure 2 (b) (second iteration). Note that the execution of the function $f(n)$ evaluates two nodes with the same value and the execution of the function L differentiates these same nodes with 3 different values.

Table 1 shows that the meta-heuristics $L1$ and $L2$ improve the performance of the heuristics $h1$ and $h2$ reducing the amount of expanded nodes. Note that $L1$ finds fewer solutions than $h1$ for solutions in levels 7 and 8. In contrast, $L2$ finds more solutions than $h2$ for that same level. It can be observed that both $L1$ and $L2$ find solutions at deeper levels than the heuristics $h1$ and $h2$. It can be said that the amount of memory required is smaller when recoding the evaluation function L to solve a set of random examples.

Table 1. Average amount of expanded nodes with quantity of solutions at different levels of the Tree. Columns 1 and 2 correspond to the results obtained when the A * strategy is applied using the heuristic h1 and the heuristic L1, respectively. Columns 3 and 4 correspond to the results obtained when using the strategy A * using the heuristic h2 and the heuristic L2, respectively

Level	h1		L1		h2		L2	
1-2	3	(119)	3	(119)	3	(119)	3	(119)
3-4	5.3054	(203)	5.0936	(203)	5	(203)	5.0	(203)
5-6	10.1255	(247)	9.0810	(247)	7.9555	(247)	7.5547	(247)
7-8	20.2105	(190)	16.2514	(183)	12.4324	(185)	10.2474	(190)
9-10	45.7760	(125)	36.1846	(130)	20.4524	(126)	16.0968	(124)
11-12	113.2812	(64)	81.2308	(65)	38.0597	(67)	31.3607	(61)
13-14	299.5217	(23)	224.9545	(22)	81.2500	(24)	90.0385	(26)
15-16			310.75	(4)	81	(3)	112,3333	(3)
17-18							0	(0)
19-20							62	(1)

The metaheuristic shows promise to complement or replace the heuristic h without impairing its properties of admissibility, as well as the generalized coordinate α (Angle of the direction of the wind with the bird) is the main cost that the bird has in its migratory journey, And has an implicit analogy with the variable g presented by the search strategy A*.

References

[1] Nayak D and B Majhi 2014 *Analysis and Implementation of Admissible Heuristics in 8 Puzzle Problem* (Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela).
[2] Tuccar M 2012 Analyzing the A* Search Heuristics with the Solutions of the 8 (University of Massachusetts Lowell).
[3] Paderborn A R 1993 Complete Solution of the Eight-Puzzle and the Bene t of Node Ordering in IDA* (Center for Parallel Computing Warburger, Paderborn, Germany).
[4] Hansson O and Mayer A and Yung M 1992 Criticizing Solutions to Relaxed Models Yields Powerful Admissible Heuristics (Elsevier Science Publishing Co).
[5] Demaine D E and Rudoy M 2017 A simple proof that the (n^2 -1) puzze is hard (MIT Computer Science and Artificial Intelligence Laboratory).
[6] Hart P and Nils J and Raphael B 1968 A Formal Basis for the Heuristic Determination of Minimum Cost (IEEE Transactions On Systems Science and Cybernetics).
[7] Wilt C and Ruml W 2015 Building a Heuristic for Greedy Search (AAAI Publications, Eighth Annual Symposium on Combinatorial Search).
[8] Bauer B 1994 The Manhattan Pair Distance Heuristic for the 15-Puzzle (Universitat-GH Paderborn, Paderborn, Warburger).
[9] Michalewicz Z and Fogel D B 2004 How to solve it: Modern heuristics (University of Massachusetts Lowell).
[10] Engelbrecht A P 2002 Computational intelligence an introduction (Center for Parallel Computing Warburger, Paderborn, Germany).
[11] Beni G and Wang J 1998 Swarm intelligence in cellular robotic systems, proceedings of the nato advanced workshop on robots and biological systems (Elsevier Science Publishing Co).

[12] Bonabeau E and Dorigo M and Theraulaz G 1999 Swarm intelligence, from natural to artificial systems (IEEE Transactions On Systems Science and Cybernetics).
[13] Beekman M and Sword G A and Simpson S J 2008 Biological foundations of swarm intelligence (AAAI Publications, Eighth Annual Symposium on Combinatorial Search).
[14] Nicoud, J and D Deneubourg, J and Floreano D 2009 Swarm intelligence in autonomous collective robotics: From tool to the analysis synthesis of distributed control strategies (Universitat-GH Paderborn, Paderborn, Warburger).
[15] Pat A and Hota A R 2011 Ant colony optimization and hypergraph covering problems (University of Massachusetts Lowell).
[16] Kobilarov M and Crane K and Desbrun M 2009 Lie group integrators for animation and control of vehicles (Center for Parallel Computing Warburger, Paderborn, Germany).
[17] Machado O M and Lima C and Gutierrez L B and Jimenez F S 2016 Uma Meta-Heurística Bio-Inspirada para um Agente de Busca.
[18] Machado O M and Lima C 2012 Predição de ajustes em vôos de aves migratórias sazonais noturnas: um simulador com adequação nebulosa (UFRJ- Rio de Janeiro- RJ).
[19] Russell and Stuart J and Peter Norvig 1995 Artificial Intelligence: A Modern Approach (Englewood Cliffs, NJ: Prentice Hall).