A Statistically Rigorous Analysis of 2D Path-Planning Algorithms

Pablo Muñoz, David F. Barrero, María D. R-Moreno

Departamento de Automática. Universidad de Alcalá
Crta. Madrid-Barcelona Km. 32,600 28871, Alcalá de Henares, Madrid, Spain
Email: (pmunoz|david|mdolores)@aut.uah.es

Path-planning is a well-known studied problem in Artificial Intelligence. Given two points in a map, path-planning algorithms search for a path that joins those two points, avoiding obstacles. It is a challenging problem with important practical applications in a wide range of applications: autonomous mobile robotics, logistics or video games, just to mention some of them. Given its importance, it has attracted much research, resulting in a large number of algorithms, some classical, such as A*, other more specialized, such as swarms. However, despite all the literature dedicated to this problem, the statistics used to analyze experimental results in most cases are naïve. In this paper we position in favor of the need of incorporating stronger statistical methods in path-planning empirical research and promote a debate in the research community. To this end we analyze some 2D-grid classical path-planning algorithms in discrete domains (i.e. A* and A* with post processing) and more recent algorithms in continuos domains (i.e. Theta* and S-Theta*). Given the differences of these algorithms, we study them under different criteria: Run-time, number of heading changes, number of expanded vertices and path-length.

Keywords: Path planning; statistic distribution; run time; heading changes; path-length

Received 12 May 2014

1. INTRODUCTION

Finding the path between two points, avoiding obstacles and using some criteria to evaluate the solution (i.e. shortest distance), has been widely studied and applied to real problems such as GPS navigation devices [1], mobile robotics [2], logistics [4] or video games [5].

In addition to the practical interest of this type of problems, it is intellectually challenging, resulting in a large number of the so called *path-planning* algorithms. The vast majority of these algorithms take a classical approximation, envisioning path-planning as a search problem abstracted by a search graph. The most common terrain discretization used for these algorithms is a regular 2D-grid with blocked and unblocked square cells [6]. For this kind of grids we can find two variants: (i) the center-node in which the mobile element is in the center of the square; and (ii) corner-node, where nodes are the vertex of the square. For both cases, a valid path is that starting from the initial node reaches the goal node without crossing a blocked cell. The most popular algorithm is A* [7] that handles a discrete state space, and its variation, A* with postprocessing (A*PS) [8] that smoothening the path obtained by A* in a post-processing step, reducing the total length. We can also mention Theta* [9] and S-Theta* [10] as algorithms specifically designed to solve path-planning algorithms that deal with continuous state space. Nonetheless, there are other approaches based on Metaheuristics such as Genetic Algorithms [11], [12], just to mention some of them.

This paper aims to promote a debate about the need of stronger statistical tools in path-planning research. The situation in path-planning research is similar to the one described by Bartz-Beielstein and Preuss in relation to the early research on Evolutionary Computation [13], based on means comparisons, and described as "paleolithic". Path-planning algorithms are typically compared by running them on different maps and then comparing the average value of the performance measure of interest among the algorithms under study. This approach, widely used in the literature, contains some methodological issues that would lead to potentially incorrect conclusions. Not even the authors of this article could have avoided these bad practices [10, 14].

From our perspective, comparison of path-planning algorithms should take benefit of proper statistical tools. We illustrate this point comparing rigorously our S-Theta* algorithm to A*, A*PS and Theta*, including strong statistical parametric tests and analyzing its scientifical significance. In preliminary research [15, 16], we studied the run-time of three path-planning algorithms (A*, Theta* and S-Theta*). In this paper we extend this work introducing the A*PS algorithm, and another two performance measures: heading changes and path-length. These two metrics are of high interest in autonomous robotics because of their impact on the energy consumption. From a methodological perspective, this paper differs from similar ones in two aspects, firstly, it tries to apply parametric statistics by fitting the statistical model to find statistical significance in the algorithms comparison. Secondly, we study the scientifical significance of the algorithms differences by means of Cohen's d statistic [17]. Finally, it is worthy to mention that the techniques described in this paper could possibly also be applied to the analysis of sampling-based path planning algorithms which are often used in robotics, such as RRT, PRM, PRM*, FMT* [18, 19].

The paper is structured as follows. First, we introduce the path-planning algorithms that are object of study, followed by a description of the research methodology used in this study. The next three sections describe the results obtained in the comparison of the path-length, heading changes and run-time of the algorithms. Finally, some conclusions are outlined.

2. PATH PLANNING

One of the first well-known algorithms to solve the shortest path problem in 2D-grids between two points is the Dijkstra algorithm [20]. Based on a distances graph, it chooses the unvisited vertex with the lowest distance, calculates the distance to each unvisited neighbor and then updates the neighbor's distance if smaller. If the destination node has been marked as visited or if the smallest tentative distance among the nodes in the unvisited set is infinity, the algorithm ends.

In order to improve the Dijkstra's algorithm performance, Hart, Nilsson and Raphael [7] proposed a new version based on heuristics named A*. This new version uses best-first search and follows a path of the lowest expected total distance calculated as a function of: the distance from the starting node to the current node x and an estimate of the distance from x to the goal. Although A* is quite fast in generating the solution, it has an important drawback, the number of nodes of the search space. It typically uses 8 neighbors nodes for the terrain discretization so it restricts the path headings to multiples of $\pi/4$, causing that A* generates a sub-optimal path with zig-zag patterns or abrupt heading changes.

In path-planning it is common to talk about the optimality of a solution only taking into consideration the path length. However, it is also interesting to include the turns required to reach the destination: an optimal solution is the smaller path between two points, so, in flat areas, this is the straight line. When there is a heading change that abandons this line, the path length grows up due to the triangle inequality. Thereby, in A* we obtain suboptimal paths as result of the restriction of the heading changes.

An A* with 16-neighbors would improve the solution smoothing the path (highly increasing the time required for the search) but would not eliminate these zig-zag patterns. The problem is that A* is designed to work with discrete state variables. In order to overcome the limitation of the discrete state space, Botea, Muller and Schaeffer [21] proposed an algorithm called A* Post Smoothed (abbreviated A*PS) to smooth the edges of A*. The idea consists of running A* and then smoothes the resulting path in a post-processing step. Therefore, the resultant path may be shorter than the original, but it increases the runtime.

Recent research in the field has proposed a new type of algorithms called "any-angle" that allows heading changes in any point of the search. For an exhaustive comparison of path-planning algorithms in continous space the reader can refer to Daniel et al. [22]. Most popular examples of this category are Field D* [23] or Theta* [9]. This last algorithm does not need a post-processing step, it does the line of sight check during the expansion of the nodes. When Theta* expands a node, p, it checks the line of sight between the parent of the node and its eight neighbors. If there is a line of sight between a successor of p and its parent, then the parent of the successor is $parent(p)$, not p like A*. When there is an obstacle blocking the line of sight, then Theta* works like A*. An alternative to Theta* is our algorithm called S-Theta* [10] that aims to reduce the amount of heading changes to reach the goal. To do this, we have based on a modified Theta* evaluation function ($F(t)$) as shown in eq. 1. The two first terms represent: $G(t)$, the cumulative cost to reach the node t from the initial node; and $H(t)$, the heuristic value, i.e, the estimated distance to the goal node. In the case of A*, the Octile heuristic is used, while in Theta* the Euclidean distance.

$$\mathrm{F}(t) = \mathrm{G}(t) + \mathrm{H}(t) + \beta(t) \tag{1}$$

The new term $\beta(t)$ added to S-Theta* gives us a measure of the deviation from the optimal trajectory to achieve the goal as a function of the direction to follow, conditional to traversing a node t. Considering an environment without obstacles, the optimal path between two points is the straight line. Therefore, any node that does not belong to that line will involve both, a change in the direction and a longer distance. Therefore, this term causes that nodes far away from that line will not be expanded during the search.

In order to compare 2D-grid path-planning algorithms[1], the length of the resulting path is usually employed as a measure of the optimality of the solution.

[1]In general, in this paper we will use the term "path-planning" to refer to path-planning in a continuous domain. Also, we will just focus on the comparison of 2D-grid algorithms.

TABLE 1: Length and heading changes values for each algorithm for a 200 X 200 grids map.

Length	A*	A*PS	Theta*	S-Theta*
20%	295	286	285	287
30%	297	288	285	291
Heading	A*	A*PS	Theta*	S-Theta*
20%	1035	209	341	131
30%	1665	531	271	348

Besides, there are other parameters such as the expanded nodes or the execution time. However, in the literature there is little work dedicated to the number of heading changes as a measure parameter, we can mention the work of Cook and Wenk [24] (or it is not properly taken into account). The optimality of the path length is highly related to the heading changes due to, as previously mentioned, the triangle inequality. Also, when heading changes are performed due to the presence of obstacles (not because of algorithm restrictions), it is also linked with the complexity of the problem, so, in any-angle algorithms, this metric will also give us some information about the algorithm performance.

Figure 1 shows an example of the solution found by each one of the algorithms under study in a map with 200 x 200 grids with 20% (right) and 30% of obstacles (left).Then, figure 2 shows the same results but representing each algorithm individually. Finally, table 1 shows the values for the lenght and the heading changes for each algorithm depending on the number of obstacles.

3. EXPERIMENTAL METHODOLOGY

Johnson [25] identified four types of experiments closely linked to a type of algorithm: application, which codes the algorithm into an application; horse race, which tries to prove that one algorithm outperforms another one; experimental analysis to better understand the algorithm and experimental average-case whose goal is to characterize an algorithm average behavior. In relation to this classification, and despite some strong and well motivated critics [26], this paper takes a mixed application and horse race approach: the goal is to provide statistical methods to compare the performance of path-planning algorithms in the context of autonomous mobile robotics. For this reason this study focuses on three performance measures of interest in this application, path-length, number of heading changes and run-time. These metrics have a great impact in battery consumption, and therefore minimizing them is highly desirable.

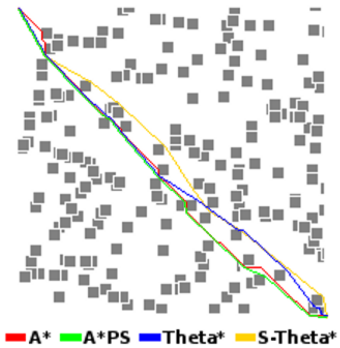
3.1. Experimental procedure

We follow the methodology suggested by Barr et al. [27]: 1) Define the goal, 2) Choose metrics, 3) Design and execute the experiment, 4) Analyze the results and provide conclusions and 5) Report. We describe them next.

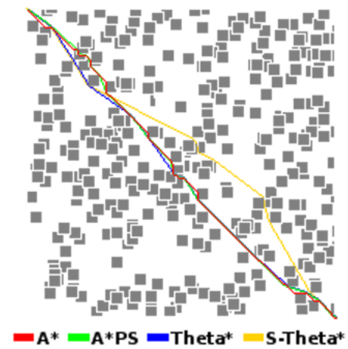
The goal of our experiments is to assess whether S-Theta* outperforms Theta* and A* outperforms A*PS in terms of path-length, path-smoothness and algorithm run-time. The metrics selected cover the previous goals and involve some of the most relevant factors in the study of path-planning algorithm. First, the runtime and number of expanded nodes gives us a measure of the complexity of the algorithm as a function of the time spent to reach a path and the memory employed in the process, respectively. Then, to analyze the optimality of the solution, we take into consideration both, the path length and the value of the heading changes. These metrics are correlated to the map hardness, so, increasing it, we can study the impact of the hardness in the performance of the algorithm and the optimality of the solution.

The design (and execution) of the experiments has to deal with which maps use. Usually path-planning research is based on a set of fixed maps that let assessing the algorithm behaviour under highly controlled problems [28]. Although these maps provide many valuable insights to the algorithms behaviour, we took a complementary approach by using a map generator. In this way we avoid two problems associated to fixed sets of maps [29]: its limited number, which could bias the conclusions, and the lack of parameterization. Of course, a map generator also introduces a bias since it needs an algorithm that will generate maps with some common properties. This undesirable effect can be handled, to some extent, applying statistical tools. In any case, map selection will bias the results, limiting the scope of the conclusions. Map parameterization is important in our study, since we are interested in analyzing how the algorithm behaviour changes under different amount of obstacles. In order to perform the statistical analysis we need several runs for each amount of obstacles, and in practice this is quite difficult to achieve with manually generated maps. We should mention that map generation introduces new variables to the experiment that are not controlled: the shape and distribution of the obstacles might have an impact on the results, but this is not studied in this work. We simply identify map hardness with amount of obstacles, which is not true in all cases. Subsection 3.2 describes map generation with detail.

The experiments involve the execution of four path-planning algorithms (A*, A*PS, S-Theta* and S-Theta*) using the same sets of maps. For the algorithms, when two nodes have the same $F(t)$ value, the tie breaking rule is to put first the node with a smaller $H(t)$, that is, the node that is, optimistic, closer to the goal. However, for Theta* and S-Theta*, the behavior is the opposite: first is the node with less $G(t)$, the node which is closer to the initial position (it



(a) 20% of obstables



(b) 30% of obstables

FIGURE 1: Overlapped paths of the 4 algorithms under study for a 200 X 200 grids map.

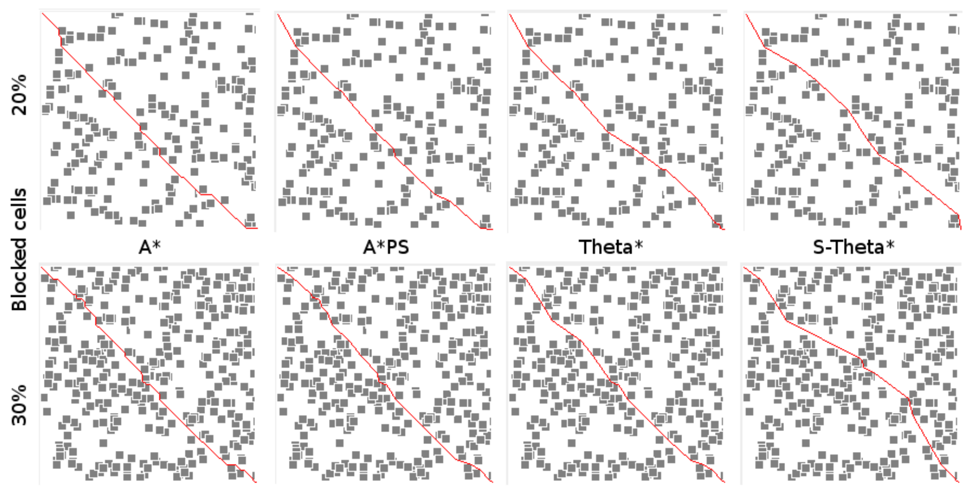


FIGURE 2: Paths of the 4 algorithms separaley with 20% & 30% of obstacles.

seems to work better as explained in [22]). Finally, the heuristic employed for all algorithms is the Euclidean distance, except for A*, which employs the octile distance. Both heuristics are valid for every algorithm, however, Euclidean distance for A* is worse than the octile distance due to the heading change restriction: the octile distance compute the distance using the movements allowed by A*. For the same reason, the other algorithms use the Euclidean distance, which allow any-angle heading.

The procedure followed in the experiments is as follows. First, we created 500 maps with 5, 10, 20, 30 and 40% percentage of obstacles, $2,500$ total maps. Then, for each subset of maps, we executed the algorithms under study (A*, A*PS, Theta* and S-Theta*), measuring the path-length, total degrees, number of heading changes and run-time. Once we got the data, we proceeded to the analysis based on statistical tests.

In order to provide a cautious study on this topic, we have used parametric statistics, which provide stronger statistical tests. Although non-parametric statistics [30] is a handy tool, we find parametric methods more convenient because, on the one hand, knowing the data model increases the test capacity of finding statistical differences; on the other hand, the statistical model might provide some clues about the nature of the data. For instance, data fitting a normal distribution, by the Central Limit Theorem, suggest that data come from an additive collection of random independent sources. In order to apply parametric statistics we need to assess whether the data satisfy the conditions. As a consequence, in first term, we have to find a good statistical distribution able to model the data, and in second term, validate the model. Once the statistical model has been found, we have used statistical tests to find significant differences in the results of the different algorithms, and complement the study with the computation of the scientifical significance. Section 3.3 motivates the need of computing the scientifical significance.

3.2. Maps generation algorithm

The Algorithm 1 generates a map with a set of obstacles randomly placed. To generate a map, a set of parameters is required to specify the dimension of the map, the dimension of the obstacles and the percentage of blocked cells. The number of blocked cells and obstacles are different (except when the obstacles have 1x1 dimension). For instance, an obstacle with a dimension of 5 columns and 4 rows blocks 20 cells.

Algorithm 1 Map generation algorithm

Require: *cols*: number of columns for the generated map
Require: *rows:* number of rows for the generated map
Require: *per*: percentage of blocked cells (< 50)
Require: *obscols*: dimension of the obstacles (columns size, ≥ 1)
Require: *obsrows*: dimension of the obstacles (rows size, ≥ 1)

```
map ← new empty map of cols ∗ rows
nblockedcells ← (per ∗ cols ∗ rows)/100
while nblockedcells > 0 do
    x ← random ∈ (0, cols)
    y ← random ∈ (0, rows)
    for ox ← x to x + obscols do
        for oy ← y to y + obsrows do
            if ox ≥ 1 and ox < cols−1 and oy ≥ 1 and oy < rows − 1 then
                if nblockedcells > 0 and map[ox][oy] not protected then
                    map[ox][oy] ← obstacle
                end if
            end if
        end for
    end for
    for sx ← x − 1 to x + obscols + 1 do
        for sy ← y − 1 to y + obsrows + 1 do
            if cx ≥ 0 and cx < cols and cy ≥ 0 and cy < rows then
                if map[cx][cy] empty then
                    map[cx][cy] ← protected
                end if
            end if
        end for
    end for
end while
return map
```

In any case, the percentage of blocked cells is computed and, if necessary, the last obstacle will be smaller to avoid a higher number of blocked cells than required. Also, to guarantee that the algorithm creates a valid map (that is, it is possible to move between every two points in the map) when an obstacle is set, its perimeter is protected to avoid superimposition of obstacles. Also, the first/last column/row are protected. Finally, as a consequence of this last one, in function of the dimension of the obstacles and the percentage of blocked cells, it is possible that the algorithm does not converge, but in practice, with less than 50% of blocked cells, the algorithm usually generates a map in a neglectable time. The pseudocode of the algorithm is in Algorithm 1 [2].

The dimension of the maps employed is 500x500 cell nodes and the percentage of obstacles (blocked or non-traversable cells) grows from 5%, 10%, up to 40%. The obstacles are generated randomly as a non-traversable block with height of 4% and width of 4% of the size of the map (for the generated maps the size of a block is 25x25). To guarantee that there exists a solution in every map, when an obstacle is added, its perimeter is protected to avoid put adjacent obstacles. Also, the first/last row and column are obstacle free, so, the initial position is the top-left corner of the map and the goal position is in the opposite side, randomly selecting a point between the bottom-right corner and the 20% of the upper nodes in the last column. In order to reduce the variance of the results, the same set of random maps were used for all the algorithms.

3.3. Comparison of path-planning performance measures

The literature on path-planning algorithms contains a large number of horse race papers. A common practice in the field is to compare the algorithms' performance just looking at the means values of the performance measure of interest. If a stochastic element is introduced in the algorithm evaluation, let's say for example a random map generator, we should expect some degree of randomness in the response and hence the performance is no longer a fixed value, but a random variable. This means that the performance measure is a composition of the algorithm performance and a stochastic component. For this reason the means comparison may induce erroneous conclusions. In addition to central tendency measures such as means, reporting should also include variability measures, or even better a sound characterization of that variability [31]. A recommended method to report variability is confidence intervals [17], which is a well known method in statistics and, in general, its computation is straightforward. The importance of proper handling of randomness has been widely discussed in related fields [32, 33, 34, 35].

Means comparison does not take into account the variability of the results, fair comparisons should include more robust statistical tools such as hypothesis testing. It provides a sound methodology to assess if data support a certain hypothesis raised by the experimenter. A common application of hypothesis testing is to determine whether two population means are equal or not, i.e., to determine statistical significance. Statistical tests are classified into two types: parametric and nonparametric tests. The former assumes that data come from a statistical

[2]The map generation algorithm and the maps used in the experiments are publicly available at `http://atc1.aut.uah.es/~david/tcj2014`

distribution, typically the normal distribution, while the latter makes no assumptions about the probability distributions of the data [30]. Parametric statistics use more information (the data probability distribution) than nonparametric statistics, and as a result it is more powerful, meaning that it is able to find statistical significance in data with smaller differences. The price to pay for this additional power is the need of finding the data probability distribution. To this end, a widely used tool is Shapiro-Wilk test [36], which tests the normality of data. Perhaps, the most widely known and used statistical test is the Students' t test, or simply t-test. It is commonly used to test if two groups of normal data come from populations with the same mean.

One problem with statistical tests is that they only find statistical significance, regardless of its magnitude or importance. Perhaps more importantly, the results of statistical tests depend on the number of samples; this is named effect size. Statistical tests provide a way to determine, for instance, if there is a significant difference between two populations, but not if that difference is relevant in practical or scientifical terms. It is possible to find very small differences just increasing the sample size, but those differences could be insignificant in practice. White [37] presents an excellent discussion about this topic and its importance in the context of experimental algorithmics.

In order to avoid the negative effects of a large sample size, and given the low cost of increasing it, assessing the scientific significance is a recommended practice. There are several metrics to evaluate the scientific significance, one of the easiest and best known ones is Cohen's d [38]. This measure is defined as the difference of the means normalized by the standard deviation:

$$d = \frac{\mu_1 - \mu_2}{\sigma'} \tag{2}$$

This definition assumes two populations with the same standard deviations. In other case, it is necessary to provide a joint standard deviation. There are different ways to perform this operation, in particular, the one that we used in this study is given by

$$\sigma' = \sqrt{\frac{\sigma_1 + \sigma_2}{2}} \tag{3}$$

which supposes that both populations have the same number of samples. The interpretation of this statistic depends on the application domain, but there is a general rule that has a wide acceptation among the statistics community setting the boundaries $d = 0.24$ as a small effect size, $d = 0.5$ medium effect size and $d = 0.74$ large effect size. Larger values would be produced by data whose difference is clearly evident without the need of sophisticated tools. Since this is a general study, not tied to any application domain, we adopt this general criteria, however any conclusion should be interpreted with care, the concept of small, medium or large effect size depends, in last instance, on the concrete application; depending on the application, small values of d might yield a large effect size.

4. STATISTICAL ANALYSIS OF PATH LENGTH

When dealing with path-planning problems, one of the most relevant characteristics of the algorithm is the path-length. The study of the path-length, that is, the distance between the initial position and the goal point without crossing a blocked cell has been a main objective because this value represents the optimality of the solution given. This distance can be always the same considering one measure (e.g. Euclidean) but we can have various (non-optimal) paths with different lengths due to the blocks cells avoidance restrictions. And actually, this is one of the elements that is considered in almost any study about path-planning algorithms.

Along this section we try to compare the path-length generated by some classical path-planning algorithms, and some non-classical algorithms such as Theta* and S-Theta* as well. In addition, we study the statistical properties of the path-length and try to find an adequate statistical model.

The main statistics in this experiment are shown in Table 2, which shows the mean, standard deviation and mean interval. We observe that the mean strongly depends on the percentage of obstacles regardless of the algorithm that was used. This is a naïve observation, since it is clear that in presence of more obstacles, the path must have more heading changes to avoid those obstacles, and it leads to an increase of its length (and computation time) due to the triangle inequality. The standard deviation has a similar behavior, larger percentages of obstacles generate higher variability of the path-length; this can be caused by the higher mean values, which can produce higher variabilities. The values shown by Table 2 suggest that S-Theta* produces paths whose lengths vary more than Theta*, A* and A*PS.

4.1. Path length statistical model fit

Figure 3 depicts the empirical probability density of the path-length obtained for different algorithms and percentage of obstacles. The distributions with 5% of obstacles have more irregularities, but the higher the percentage of obstacles, the smoother the distribution is. Additionally, Figure 3 represents overlapped to the empirical distribution a normal distribution whose parameters have been fitted using maximum-likelihood, and it can be observed that easy maps produce path-length with a worse fit than hard maps.

The results of the Shapiro-Wilk test are shown in Table 3. Those cases where data does not support the normality hypothesis have been marked according to

TABLE 2: Summary of the path-length, including mean, standard deviation, median, and 95% normal confidence intervals for the mean. Each combination of algorithm and percentage of obstacles were repeated with 500 maps. The lowest mean path-length for each number of obstacles is marked with bold. To better visualize tendencies in these data, reader should check out Figure 3.

Obstacles	Algorithm	Mean	Sd	Median	Mean interval
5%	A*	691.1	16.8	687.5	[690, 693]
5%	A*PS	675.8	19.3	673.9	[674, 678]
5%	Basic Theta*	**675.0**	19.5	673.3	[673, 677]
5%	S-Theta*	677.8	20.4	675.6	[676, 680]
10%	A*	696.6	19.5	694.2	[695, 698]
10%	A*PS	678.3	19.8	677.7	[677, 680]
10%	Basic Theta*	**676.9**	19.7	675.7	[675, 679]
10%	S-Theta*	682.8	22.4	681.8	[681, 685]
20%	A*	711.1	23.4	711.9	[709, 713]
20%	A*PS	688.2	22.4	687.6	[686, 690]
20%	Basic Theta*	**685.6**	21.6	685.5	[684, 688]
20%	S-Theta*	698.7	28.2	698.1	[696, 701]
30%	A*	728.5	25.2	728.7	[726, 731]
30%	A*PS	702.6	23.9	703.2	[700, 705]
30%	Basic Theta*	**698.9**	23.0	698.9	[697, 701]
30%	S-Theta*	721.2	33.9	717.9	[718, 724]
40%	A*	746.2	26.1	746.0	[744, 748]
40%	A*PS	718.7	25.4	718.2	[716, 721]
40%	Basic Theta*	**713.9**	24.2	713.9	[712, 716]
40%	S-Theta*	741.3	33.8	739.9	[738, 744]

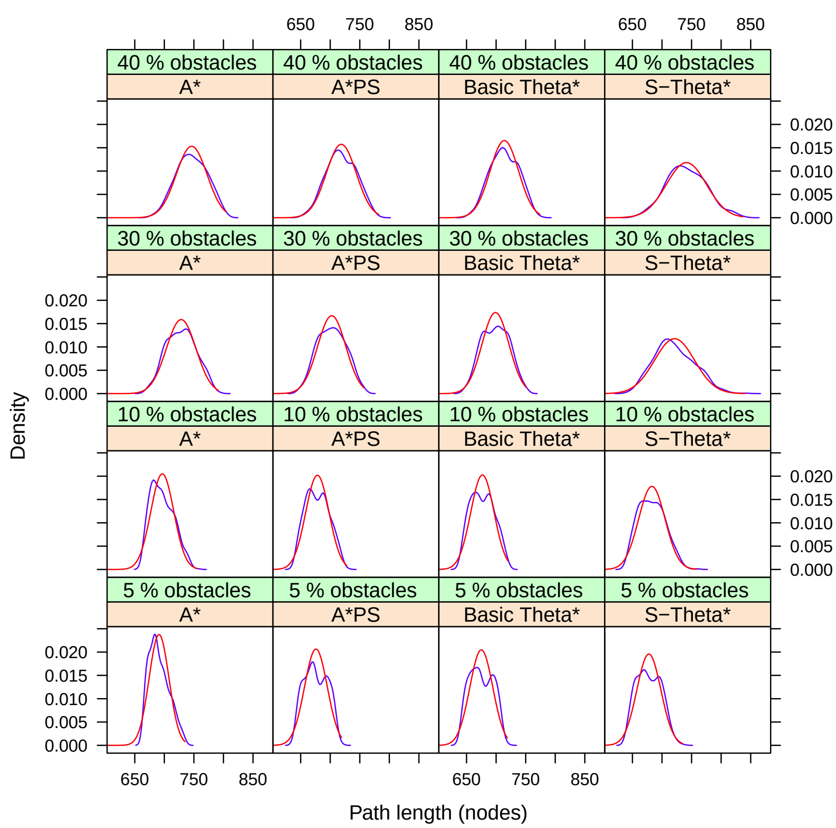


FIGURE 3: Empirical probability density function of the pathLength (red) compared to a normal distribution fitted using maximum-likelihood (blue). Each column represents data from the same algorithm (A*, A*PS, Theta* or S-Theta*) while rows correspond to the same amount of obstacles.

TABLE 3: Shapiro-Wilk test of normality of the path-length generated by A*, A*PS, Theta* and S-Theta*. Those p-values that show evidence to reject the normality hypothesis have been marked depending on the significance level, • when $\alpha = 0.05$, •• when $\alpha = 0.01$ and • • • when $\alpha = 0.001$.

Obstacles	P-value A*	P-value A*PS	P-value Theta*	P-value S-Theta*
5%	0.118	0.019	0.001 • • •	0.006 ••
10%	0.032 •	0.178	0.004 ••	0.104
20%	0.471	0.083	0.025 •	0.427
30%	0.956	0.884	0.241	0.219
40%	0.284	0.871	0.145	0.690

TABLE 4: T-test of the mean difference between the path length produced by A* and A*PS algorithms with different presence of obstacles. T-statistic, confidence interval (CI) with $\alpha = 0.05$ for the difference of means, t-test and p-value are reported. In addition, the table also shows Cohen's d statistic (D) for scientific significance. All the cases provided strong statistical evidence with very low p-values and large effect sizes.

Obstacles	T	CI	P-value	D
5%	5.59	[8.94, 18.69]	p≪0.01	0.79
10%	4.85	[7.80, 18.50]	p≪0.01	0.69
20%	8.74	[20.84, 32.99]	p≪0.01	1.24
30%	8.20	[21.29, 34.77]	p≪0.01	1.16
40%	8.21	[22.32, 36.45]	p≪0.01	1.16

the confidence level[3]. The results suggest that, in effect, algorithms, at least to some extent, tend to generate normal path-lengths. There are also differences among the algorithms, 3 out of 5 cases of Theta* path-length normality tests provided evidence to reject the normality, while we cannot reject normality in any case for A*PS.

Looking at these results and Figure 3 we can conclude that the algorithm and the percentage of obstacles influence the normality of the path-length. Higher percentage of obstacles tend to generate more normal path-lengths, while some algorithms (A*/A*PS) tend also to generate more normal path-lengths than others (Theta*/S-Theta*). In any case, with these results, it seems reasonable to assume the normality of the path-length when the percentage of obstacles is higher than 20%. As a consequence of this evidence, we found support to apply normal-distribution based statistics to compare the path-length generated by the algorithms, and in particular we can apply in the case of two algorithms classical t-tests.

TABLE 5: T-test of the mean difference between the path length produced by Theta* and S-Theta* algorithms with different presence of obstacles. T-statistic, confidence interval with $\alpha = 0.05$ for the difference of means, t-test and p-value. In addition, the table also shows Cohen's d statistic (D) for scientific significance. All the cases but 5% obstacles provided strong statistical evidence with very low p-values and effect sizes from medium ($d = 0.41$) to large ($d > 0.9$).

Obstacles	t	CI	p-value	D
5%	-0.03	[-5.39, 5.23]	0.98	-0.00
10%	2.93	[2.76, 14.10]	p≪0.01	0.41
20%	4.00	[6.92, 20.41]	p≪0.01	0.57
30%	8.71	[27.72, 43.96]	p≪0.01	1.23
40%	6.48	[18.44, 34.57]	p≪0.01	0.92

4.2. Comparison of path-lengths

We first looked for statistical significance between A* and A*PS, the main results can be seen in Table 4. The table shows the t statistic, the confidence interval with $\alpha = 0.05$ of the difference of the means, the p-value of the t-test and finally the d statistic for scientific significance. All the tests, regardless of the percentage of obstacles, provided a large statistical significance with p-values much smaller than 0.01, so we reject the null hypothesis of no difference. In addition to the statistical significance, we find that Cohens' d achieve large values, between 0.69 and 1.24, which means that the difference of means is between half and more than one sigmas. The high scientific significance found in the data is supported by the confidence intervals, 0 falls far from all the intervals. So, data supports the conclusion that A*PS outperforms A* generating shorter paths.

The second test looked for differences between the path-length means generated by Theta* and S-Theta*. The result is summarized in 5. It shows the t-statistic, confidence interval of the difference of the means, p-value and Cohen's d of the path-length of both Theta algorithms. It can be seen that there is no evidence

[3] We use • instead of the standard * symbol in the literature to avoid confusion with the algorithms' name.

to reject the null hypothesis when there are 5% of obstacles, and actually, this case did not pass the normality test, so, we cannot use this result to depict any sound concussion. On the contrary, the other cases provided again very low p-values with medium to large d, so we can reject the mean equality hypothesis.

In this section we have provided strong evidence showing that the analyzed path-planning algorithms generate path-length that follow a normal distribution. The normality of the path-length is more clear when the percentage of obstacles is higher, when it is around 5% this claim has no support. Using this result, the path-length generated by A* and A*PS were compared looking for statistically and scientifically differences, and the results actually suggest that there is a difference. The same can be said about Theta* and S-Theta* for maps with a percentage of obstacles higher than 5%.

5. STATISTICAL ANALYSIS OF HEADING CHANGES

One application domain where path-planning algorithms have been widely used is autonomous mobile robots. One of the most important constrains in this domain is the robot battery, which is a limited resource and usually its usage must be optimized as much as possible. The path-planning algorithm has a big impact on the battery usage: short paths, under the same conditions, require less battery usage than longer paths. There is, however, another element in the planned path with a major impact on energy consumption, the heading changes. When a robot changes its direction, it must activate some additional motion subsystems to move, for instance, the direction of its wheels (in case it was a rover), legs (in case it was a walker) or any other mechanism, action that is expensive from the energy perspective. So, the number of heading changes should be minimized. In this section we provide an empirical study about the behavior of the four algorithms under study from the point of view of the curvature of the planned path. Smoother paths with few curves are in general preferred.

In order to characterize the algorithm from this perspective, we have used two measures: *Heading changes* and *total degrees*. The former is the number of direction changes made by the path, while the latter is the sum of the angles that the path changes. These two measures are closely related to each other, as later will be shown. We are interested in analyzing the algorithm behavior from this point of view, and, in particular, to verify if S-Theta* is able to generate smoother paths than the original Theta* algorithm.

The heading changes metric is quite complex to analyze: first, its importance is relative to the tradeoff between the path length and the requirements. In some applications the cost of a heading change can be zero while in other applications it is very relevant. Also, the heading changes can vary as function of the geometry of the map. Maps with randomly separated obstacles could lead in high number of heading changes meanwhile, in maps with big obstacles or with symmetric distribution, may be possible that a few heading changes are made but with bigger amplitude. For this reason, we also include the total degrees as a metric, which is similar to the path length. The total degrees are more independent of the obstacles distribution, but as the path length, highly correlated with it.

A summary of the results can be found in Tables 6 and 7. Some important statistics are shown, including the mean, standard deviation and mean intervale. An overview to the tables shows the dependence of the path curvature with the presence of obstacles: the number of curves in the paths increases with the number of obstacles; this correlation also holds for the variability of the measures for all the algorithms. A* deserves a special mention, it generates paths with heading changes and total degrees means much larger than other algorithms. The reason of this behavior is the restriction of the heading changes: to avoid obstacles A* requires turns of $\pi/4$ while the other algorithms can turn an arbitrary angle.

When there is a small number of obstacles ($\lesssim$10%) the algorithms exhibit a deterministic behavior. This is due to how they search the path: a heuristic guides the search towards the goal position, thus, the algorithm follows the optimistic route until an obstacle forces a heading change. At this point, the search starts expanding nodes which will not be analyzed if there were no obstacle, thus, with every heading change, the algorithm needs to expand some new nodes. When there are only a few obstacles, the algorithm usually does not need to change its direction, and thus, usually follows a close to optimal path.

Prior to continue studying the statistical model of the data, it is worth to observe how heading changes and total degrees relate to each other to simplify the analysis. These results can be seen in Table 8, that quantifies the correlation coefficients. In all cases, the correlation coefficients are quite large, regardless of the algorithm or percentage of obstacles; this is evident in the case of A*, with correlation coefficients higher than 0.9. Table 8 also suggests stronger correlations in maps with more obstacles. This strong correlation is not surprising given the direct relationship between both variables: as the number of heading changes increases, the total degrees also increases. As a consequence of the correlation, we can simplify the study excluding one of these statistics without a significant loss of information. We have used this fact to select the measure that offers better properties to perform the statistical analysis, i.e., the one with best statistical model fit.

Tables 9 and 10 show the result of the normality tests for both measures, heading changes and total degrees. Given the clear dependence of the data with the number

TABLE 6: Summary of *total degrees*, including mean, standard deviation, median and 95% normal confidence intervals for the mean. Each combination of algorithm and percentage of obstacles were repeated with 500 maps. The lowest mean total degrees for each number of obstacles is marked with bold.

Obstacles	Algorithm	Mean	Sd	Median	Mean interval
5%	A*	326.2	205.2	270	[308, 344]
5%	A*PS	22.0	36.0	9	[19, 25]
5%	Basic Theta*	**15.6**	25.3	7	[13, 18]
5%	S-Theta*	20.9	21.7	14	[19, 23]
10%	A*	442.9	333.7	360	[414, 472]
10%	A*PS	50.0	55.4	31	[45, 55]
10%	Basic Theta*	39.7	47.5	23	[36, 44]
10%	S-Theta*	**37.4**	28.7	33	[35, 40]
20%	A*	695.8	455.6	585	[656, 736]
20%	A*PS	133.0	98.0	108	[124, 142]
20%	Basic Theta*	111.5	90.1	86	[104, 119]
20%	S-Theta*	**81.5**	45.6	70	[78, 86]
30%	A*	914.8	444.1	855	[876, 954]
30%	A*PS	253.4	140.0	242	[241, 266]
30%	Basic Theta*	216.8	129.7	193	[205, 228]
30%	S-Theta*	**149.2**	77.0	135	[142, 156]
40%	A*	1148.4	432.6	1035	[1110, 1186]
40%	A*PS	407.6	175.1	397	[392, 423]
40%	Basic Theta*	358.8	161.2	341	[345, 373]
40%	S-Theta*	**262.4**	133.0	237	[251, 274]

TABLE 7: Summary of the number of *heading changes*, including mean, standard deviation, median and 95% normal confidence intervals for the mean. Each combination of algorithm and percentage of obstacles were repeated with 500 maps. The lowest mean heading changes for each number of obstacles is marked with bold.

Obstacles	Algorithm	Mean	Sd	Median	Mean interval
5%	A*	7.21	4.57	6	[6.81, 7.61]
5%	A*PS	1.28	1.03	1	[1.19, 1.37]
5%	Basic Theta*	1.03	0.90	1	[0.95, 1.11]
5%	S-Theta*	**0.90**	0.74	1	[0.83, 0.96]
10%	A*	9.75	7.41	8	[9.10, 10.40]
10%	A*PS	2.47	1.41	2	[2.34, 2.59]
10%	Basic Theta*	2.07	1.31	2	[1.95, 2.18]
10%	S-Theta*	**1.60**	0.94	2	[1.52, 1.68]
20%	A*	15.31	10.12	13	[14.42, 16.19]
20%	A*PS	4.92	1.95	5	[4.74, 5.09]
20%	Basic Theta*	4.18	1.90	4	[4.01, 4.34]
20%	S-Theta*	**3.09**	1.28	3	[2.98, 3.20]
30%	A*	20.06	9.86	18	[19.20, 20.93]
30%	A*PS	7.70	2.38	8	[7.49, 7.91]
30%	Basic Theta*	6.69	2.34	6	[6.48, 6.90]
30%	S-Theta*	**4.83**	1.69	5	[4.69, 4.98]
40%	A*	25.14	9.61	23	[24.30, 25.99]
40%	A*PS	10.93	2.95	11	[10.67, 11.19]
40%	Basic Theta*	9.74	2.82	10	[9.49, 9.99]
40%	S-Theta*	**7.26**	2.46	7	[7.04, 7.47]

of obstacles, we have performed different normality tests varying the percentage of obstacles. The p-values of A* tests show evidences to reject the normality hypothesis, the same can be said for almost any algorithm with 5% and 10% obstacles. On the contrary, we did not find evidence to reject the normality of total degrees, neither heading changes for any algorithm but A*, and percentage of obstacles higher than 20%. For this reason, we exclude those runs with 5% and 10% from the hypothesis testing.

Given the normality test in Table 10, it seems that heading changes with at least 20% of obstacles

TABLE 9: Shapiro-Wilk normality test for total degrees. Rejected normality tests with $\alpha = 0.01$ are marked with "•".

Obstacles	P-value A*	P-value A*PS	P-value Theta*	P-value S-Theta*
5%	p≪0.01 •	p≪0.01 •	p≪0.01 •	p≪0.01 •
10%	p≪0.01 •	p≪0.01 •	p≪0.01 •	0.187
20%	p≪0.01 •	0.007 •	0.002 •	0.062
30%	p≪0.01 •	0.016	0.001 •	0.087
40%	p≪0.01 •	0.338	0.674	0.003 •

TABLE 10: Shapiro-Wilk normality test for heading changes. Rejected normality tests with $\alpha = 0.01$ are marked with "•".

Obstacles	P-value A*	P-value A*PS	P-value Theta*	P-value S-Theta*
5%	p≪0.01 •	p≪0.01 •	p≪0.01 •	p≪0.01 •
10%	p≪0.01 •	0.001	p≪0.01 •	p≪0.01 •
20%	p≪0.01 •	0.142	0.205	0.004 •
30%	p≪0.01 •	0.028	0.053	0.011
40%	p≪0.01 •	0.182	0.339	0.014

TABLE 8: Total degrees and heading changes correlation coefficients. Each combination of algorithm and percentage of obstacles were repeated with 500 maps.

Algorithm	Obstacles (%)	Correlation
A*	5	0.9989
A*	10	0.9991
A*	20	0.9993
A*	30	0.9987
A*	40	0.9981
A*PS	5	0.7235
A*PS	10	0.7996
A*PS	20	0.8210
A*PS	30	0.8544
A*PS	40	0.8988
Basic Theta*	5	0.7084
Basic Theta*	10	0.8014
Basic Theta*	20	0.8494
Basic Theta*	30	0.8790
Basic Theta*	40	0.8910
S-Theta*	5	0.7646
S-Theta*	10	0.7611
S-Theta*	20	0.7632
S-Theta*	30	0.7884
S-Theta*	40	0.8603

follow a normal distribution. For this reason we have selected this measure to compare whether S-Theta* generates smoother paths than Theta*; the normality allows us to perform a simple t-test. In order to avoid detecting nonrelevant differences, we have also computed the Cohen's d statistic to assess the scientific significance of the mean differences. The result is shown in Table 11, which only includes those obstacles

TABLE 11: T-test for the difference of heading changes means resulting from Theta* and S-Theta*. The table reports the t statistic, confidence interval (CI) and p-value. In addition, the table also shows Cohen's d statistic (D) for scientific significance.

Obstacles	T	CI	P-value	D
20%	-3.97	[-1.36, -0.46]	p≪0.01 •	-0.56
30%	-6.53	[-2.49, -1.33]	p≪0.01 •	-0.92
40%	-5.16	[-2.78, -1.24]	p≪0.01 •	-0.73

that showed normality. The p-values have very low values, suggesting that the mean heading changes given by Theta* and S-Theta* are, actually, different. The confidence interval and the d statistic also suggest that the effect size is relevant, between 0.5 and 0.9 sigmas.

6. STATISTICAL ANALYSIS OF RUN-TIME

We have selected the execution time to measure run-times. In order to make a fair comparison, the implementation of the three algorithms use the same methods and data structures to manage the information grid[4]. The measurement of the run-time by the CPU time has several drawbacks [39], but in this case we think it is justified because the algorithms contains computations that are not well captured by machine-independent time measures, such as the number of expanded nodes. Of course, the price we have to pay is an increased difficulty to repeat and compare these results.

Table 12 summarizes some statistics of the measured

[4]The execution was done on a 2 GHz Intel Core i7 with 4 GB of RAM under Ubuntu 10.10 (64 bits)

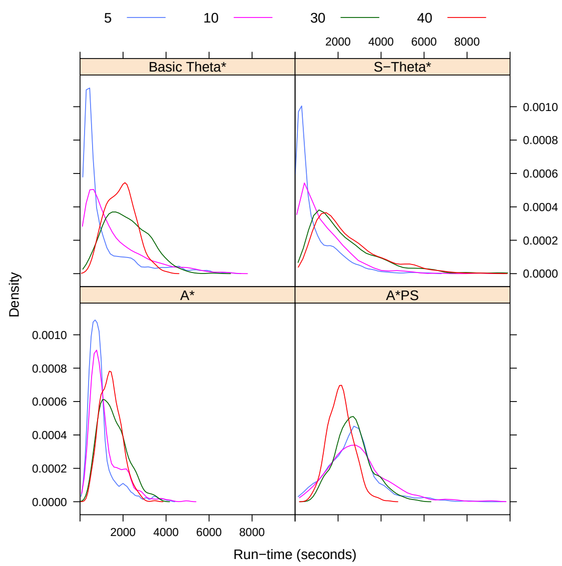


FIGURE 4: Empirical density probability function of the run-time for the four algorithms under study with different map difficulties.

run-times. For clarity, we have depicted the empirical density probability function of the run-time in Figure 4, the diagrams were grouped by algorithm and presence of obstacles in the map.

We observe in Figure 4 and Table 12 that the shape of the run-time varies as a function of the algorithm and the amount of obstacles. The run-times of A*/A*PS present a smaller variance and seems to be more symmetrical than the other two algorithms, and it increases with the problem hardness. On the contrary, the distribution of the run-time for Theta*/S-Theta* algorithms has a long right tail regardless the amount of obstacles in the map. Surprisingly, the presence of obstacles seems not to affect the run-time of the algorithm, one might expect hard maps associated to longer run-times. In the case of A*/A*PS even the shape and variance of the distributions seem to be unaffected by the presence of obstacles. The nature of A*/A*PS explains these observations, A*/A*PS expand more nodes, however, the operations performed on each node are the same, yielding a more deterministic run-time. On the contrary, Theta*/S-Theta* require the computation of the line of sight, whose computation grows exponentially. The run-time required to solve hard problems has more variance than easy problems, but the difference is quite small.

Except for S-Theta*, we can observe that the runtime has a particular behavior in function of the percentage of obstacles: the runtime trends to be directly proportional to the number of obstacles until one point in which the proportionality is inverted and the runtime goes down when more obstacles are added to the maps. This is due to the nodes expanded during the search. When there are a great number of obstacles, the algorithms have fewer choices for the path, and thus, there are less nodes that can be analyzed.

Also, for any-angle algorithms, there is an important component inside the algorithm: the line of sight check. This computation has exponential complexity in function of the distance between the two nodes to connect. When greater is the distance between two nodes, more nodes shall be checked. In maps with a small number of obstacles, the line of sight check usually is computed for distant nodes, meanwhile in maps with increasing number of obstacles, the line of sight check will be more often interrupted due to the presence of an obstacle, reducing the time consumption in that task.

Although the previous explanation applies to S-Theta*, its runtime is directly proportional to the number of obstacles in any case. S-Theta* has a particularity that is not present in Theta*: the last one only made heading changes at the corner of the obstacles, while our algorithm does not. S-Theta* tries to minimize the heading changes and thus, it is possible that the heading change can be produced at any point of the map if it smoothes the final solution, performing fewer heading changes as previous results shown. For this reason, usually S-Theta* computes longer line of sight checks than Theta* in maps with more obstacles. Also, the complexity in the node expansion is higher in S-Theta* due to the angles computation, which require floating point operations.

With the results reported in the literature, we have tried to fit data to the three distributions: normal, lognormal and Weibull. Literature on run-time analysis often mention the exponential distribution instead of the normal, but looking at Figure 4 it seems clear that data do not fit an exponential distribution, at least when the maps contain more than 20% of obstacles.

The diagrams in Figure 5 are depicted with a set of overlapping distributions, which correspond to the three statistical distributions previously mentioned. The parameters of these distributions were fitted by maximum likelihood. As the reader can observe in Figure 4, the distributions that better fit our data in most cases are the lognormal and normal. The exceptions to this observation are A*PS, Theta* and S-Theta* solving maps with a very low number of obstacles; in those cases data exhibit some irregularities that are badly described by any distribution. Curiously, A* presents a pronounced peak which is not well described by the distribution. Even in this case, the A* run-time is well described by the lognormal and normal distributions. On the contrary, the Weibull does not seem to describe well any case.

Previous studies on run-time analysis have identified the lognormal distribution as a good model. In this case, lognormal and normal distributions seem to be quite close to each other, and actually, it is not possible to find by visual inspection big differences. This fact can be explained by the well-known convergence of these distributions, helped by the relatively large mean value

TABLE 12: Summary of total run-time, including mean, median, standard deviation (sd) and 95% normal confidence intervals for the mean. Each combination of algorithm and percentage of obstacles were repeated with 500 maps. The lowest mean heading changes for each number of obstacles is marked with bold. To better visualize tendencies in these data, reader should check out Figure 4.

Obstacles	Algorithm	Mean	Sd	Median	Mean interval
5%	A*	947	630	784	[892, 1003]
5%	A*PS	2709	1260	2692	[2598, 2819]
5%	Basic Theta*	1116	1313	468	[1001, 1231]
5%	S-Theta*	**867**	976	383	[782, 953]
10%	A*	**1147**	726	923	[1083, 1211]
10%	A*PS	2921	1380	2772	[2799, 3042]
10%	Basic Theta*	1542	1355	1087	[1423, 1661]
10%	S-Theta*	1332	1433	934	[1206, 1458]
20%	A*	**1445**	764	1280	[1378, 1513]
20%	A*PS	3053	1210	2954	[2947, 3160]
20%	Basic Theta*	1997	1258	1702	[1887, 2108]
20%	S-Theta*	1770	1512	1359	[1637, 1903]
30%	A*	**1541**	660	1428	[1483, 1599]
30%	A*PS	2718	862	2660	[2642, 2794]
30%	Basic Theta*	2185	1004	2060	[2097, 2273]
30%	S-Theta*	2258	1597	1758	[2117, 2398]
40%	A*	**1424**	500	1400	[1380, 1468]
40%	A*PS	2122	584	2102	[2070, 2173]
40%	Basic Theta*	1897	686	1897	[1837, 1958]
40%	S-Theta*	2419	1505	1971	[2287, 2551]

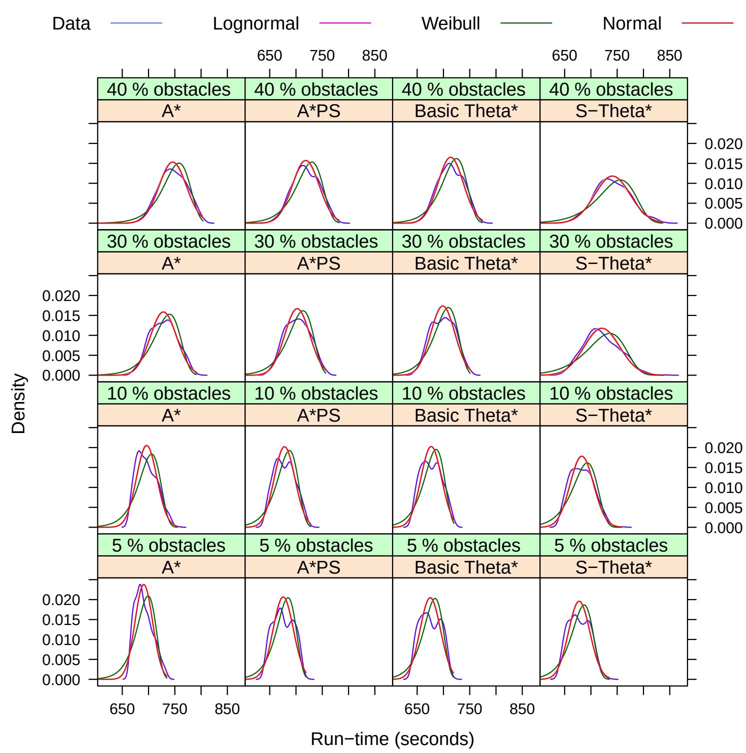


FIGURE 5: Empirical density function of the run-time overlapped to three distributions: Lognormal, Weibull and normal.

and small skewness of the data.

Following the run-time literature, and the fact that the normal distribution gives a non-zero probability to negative run-times [40], the lognormal distribution seems a better choice to model the run-time. With this in mind, and the previous result, it is reasonable to focus the study on the lognormal distribution and hypothesize that the run-time of the studied algorithms applied to path-planning problems follow a lognormal distribution.

One of the most interesting properties of the lognormal distribution is its close relationship to the normal distribution, actually, lognormal data can be converted into normal data by simply taking logarithms [40]. We use this property to assess the lognormality of the run-time in a more formal way, just applying the Shapiro-Wilk test of normality to the logarithm of the run-time, whose result is shown in Table 13.

It is always possible to increase the sensibility of a statistical test by means of increasing the sample size [37], which might lead to find statistical significance with arbitrary small (meaningless) differences; in other words, the results of the statistical tests depend on the sample size. This issue is relevant in computational science, since generating samples may be inexpensive. Following Ridge's recommendations [41], we have computed the tests using 30 random runs. In addition, we computed Cohen's d that, as described in section 3, measures the difference size and is invariant with the sample size.

The results confirm our previous observations about the lognormality of the run-time. The p-values shown in Table 13 in almost all the cases are quite high, which means that the hypothesis of lognormality is compatible, with a high probability, with our data. However, when the amount of obstacles is low, 5%, only A* passed the test with $\alpha = 0.01$, so we reject the lognormality of run-times of A*PS, Theta* and S-Theta* with $\alpha = 0.01$. In addition, the p-value of A*PS with 20% obstacles takes 0.012, so there is evidente to reject the lognormality of this random variable with $\alpha = 0.05$. Since those run-times with 5% obstacles do not exhibit lognormal behaviour, we exclude them from the following study.

Looking at the median and standard deviation of A* and A*PS shown Table 12, we observe that there is a large difference in their values. For example, with 40% obstacles, the median of A* is 1424 while A*PS is 2122, both with a standard deviation around 550. These values suggest a significant difference. If we take into account that the design of A*PS determines that it introduces additional computations in relation to A*, it seems reasonable to conclude that the run-time of A*PS is higher than A* without a statistical test.

The case of Theta* and S-Theta* is not so clear. Their medians and variances are close enough to need a quantitative analysis. Once we have found a good statistical model, we can use parametric statistics to compare the run-times of S-Theta* and Theta*. We have assessed the statistical and scientific significance of the mean run-time differences. The statistical significance was computed with a simple t-test of the logarithm of the run-times, finding no statistical significance between the run-time means of Theta* and S-Theta*. Table 14 summarizes the result of the t-test, including the confidence interval of the mean differences and Cohen's d statistic.

TABLE 14: T-test for the mean difference of the run-time of Theta* and S-Theta*. The table reports the t statistic, confidence interval (CI) of the mean difference, p-value. In addition, the table also shows Cohen's d statistic (D) for scientific significance.

Obstacles	T	CI	P-value	D
10%	-0.96	[-0.37, 0.13]	0.34	-0.14
20%	-1.64	[-0.40, 0.04]	0.10	-0.23
30%	-0.91	[-0.26, 0.10]	0.37	-0.13
40%	1.37	[-0.04, 0.23]	0.17	0.19

TABLE 15: T-test for the mean difference of the run-time of Theta* and S-Theta* using 500 samples. Compare these values to the ones shown in Table 14, which are computed with 100 samples. The table reports the t statistic, confidence interval (CI) of the mean difference, p-value and Cohen's d.

Obstacles	T	CI	P-value	D
10%	-3.62	[-0.32, -0.09]	p≪0.001 •••	-0.23
20%	-4.08	[-0.28, -0.10]	p≪0.001 •••	-0.26
30%	-1.83	[-0.15, 0.00]	0.07	-0.12
40%	4.12	[0.07, 0.20]	p≪0.001 •••	0.26

7. DISCUSSION ABOUT THE EFFECT SIZE

It is important to take into account the effect size and the scientifical significance of the result. In order to illustrate this, we have repeated the experiment summarized in Table 14 with one notable diference: instead of just including 100 samples, we used all the 500 samples and performed the t-test. The result of this experiment is in Table 15.

The only difference between the tests reported in 14 and 15 is the sample size. However, the conclusions induced by both tests are radically different. Three out of four cases in Table 15 show statistical difference, it contrasts with the test done with 100 samples. The sensibility of the test to find differences increases with the number of samples, to a point that the test is able to find significance with very small diferences. On the contrary than p-values, Cohen's d remains robust with the increase of the samples. Its value has changed, but in much less magnitude than p-values, i.e., d is more robust than p-values. Even though the t-test found statistical difference, Cohen's d values around 0.2 add an important insight to this picture: According to the criteria introduced in section 3.3, the scientifical

TABLE 13: Shapiro-Wilk test of normality of the logarithm of the run-time. With these p-values we cannot reject the hypothesis of its normality, which means that we cannot reject the lognormality of the run-time. Only Theta* and S-Theta* with a ratio of 5% of obstacles provide evidence to let us reject the normality of the run-time. Rejected normality tests are marked with • when $\alpha = 0.05$, •• when $\alpha = 0.01$ and ••• when $\alpha = 0.001$.

Obstacles	P-value A*	P-value A*PS	P-value Theta*	P-value S-Theta*
5%	0.747	p<0.001 •••	0.008 ••	p<0.001 •••
10%	0.274	0.087	0.114	0.097
20%	0.847	0.012 •	0.149	0.743
30%	0.079	0.057	0.064	0.897
40%	0.224	0.540	0.119	0.972

significance of the test is quite low. Therefore, even if there is statistical difference, in practical terms, that difference is irrelevant.

Usually, path-planning algorithms are compared in the literature running them to solve a collection of problems. The analysis of the results done in the literature uses to be quite naïve, the common practice is just computing the mean run-time for each parameter in the experiment. So, when they conclude that an algorithm is better than another, they only compare the average results of multiples executions. This value does not show much information: in the case of the runtime, this value depends on the complexity of both, the algorithm and the map. So the average results are translated in a loss of information that could be useful to compare and characterize the algorithms. This mistake is critical when the parameters have a big dispersion. If we avoid using the average values we can obtain a way to compare more precisely the algorithms.

8. CONCLUSIONS

In this paper we have tried to open a discussion in the path-planning research community about the need of integrating stronger statistical methods. In addition, we have analyzed our algorithm S-Theta* in relation to its forerunner, Theta*, and two classical path-planning algorithms: A* and A* with postprocessing (A*PS). The design of S-Theta* considers the application domain of autonomous robotics, and therefore one of its main objectives is to minimize energy consumption. With this consideration, the paper has analyzed two traditional performance measures, the path-length and run-time, along with the number of heading changes and the total degrees.

This paper followed an empirical approach. We have run the algorithms to plan paths in maps with a varying number of obstacles (and therefore difficulties) and measured the path-length, run-time, heading changes and total degrees. We have tried to fit a statistical model and performed parametric tests to find statistical significance. In addition, we have computed statistics to determine the scientifical significance of the results.

In our experiments, the path-length of the algorithms seems to be well modelled by the normal distribution. There are, however, some exceptions when the number of obstacles is low; Theta* needs a high percentage of obstacles in the map (30-40%) to generate normally distributed paths. We have also compared the path-length generated by A* with the ones generated by A*PS, and Theta* with S-Theta*. The results of the parametric tests showed that A* outperforms A*PS under this criteria while Theta* outperforms S-Theta*. We have found statistical and scientifical significance.

The number of heading changes and total degrees showed to be highly correlated, and we have used this to simplify the study just by considering one of them. We have selected the number of heading changes because it fits better a normal distribution. As in the previous metric, heading changes exhibit normal distribution for maps with medium and high obstacles percentage, so A* and A*PS were excluded from the test. This test has shown statistical and scientifical difference between Theta* and S-Theta*, and the data suggest that S-Theta* produces paths with less heading changes in comparison to Theta*. In relation to run-time, unless the number of obstacles is very low (5%), it seems to follow a lognormal distribution. However, we did not find statistical difference for Theta* and S-Theta*.

ACKNOWLEDGEMENTS

Pablo Muñoz is supported by the European Space Agency (ESA) under the Networking and Partnering Initiative (NPI) *Cooperative Systems for Autonomous Exploration Missions*.

REFERENCES

[1] Kanoulas, E., Du, Y., Xia, T., and Zhang, D. (2006) Finding fastest paths on a road network with speed patterns. *Proceedings of the 22nd International Conference on Data Engineering, 2006*, Atlanta, Georgia, USA, 3-7 April.

[2] Muñoz, P., R-Moreno, M., Martinez, A., and Castaño, B. (2012) Fast path-planning algorithms for future mars exploration. *Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS'12)*, Turin, Italy, 4-7 September.

[3] Hachour, O. (2008) Path planning of autonomous

mobile robot. *International Journal of Systems, Applications, Engineering & Development*, **2**, 178–190.

[4] Bello-Orgaz, G., Barrero, D., R-Moreno, M., and Camacho, D. (2014) Acquisition of business intelligence from human experience in route planning. *Enterprise Information Systems*, **to be published**.

[5] Millington, I. and Funge, J. (2009) Artificial intelligence for games, chapter. Pathfinding. Morgan Kaufmann Publishers.

[6] Yap, P. (2002) Grid-based path-finding. *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, **2338**, pp. 44–55. Springer Berlin / Heidelberg.

[7] Hart, P., Nilsson, N., and Raphael, B. (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics.*, **4**, 100–107.

[8] Botea, A., Muller, M., and Schaeffer, J. (2004) Near optimal hierarchical path-finding. *Journal of Game Development*, **1**, 1–22.

[9] Nash, A., Daniel, K., Koenig, S., and Felner, A. (2007) Theta*: Any-angle path planning on grids. *In Proceedings of 22nd National Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, 22-26 July, pp. 1177–1183.

[10] Muñoz, P., Barrero, D. F., and R-Moreno, M. (2012) S-Theta*: low steering path-planning algorithm. *Procs. of the Thirty-second SGAI International Conference on Artificial Intelligence*, Cambridge, UK, 10-12 December.

[11] Elshamli, A., Abdullah, H., and Areibi, S. (2004) Genetic algorithm for dynamic path planning. *Proceedings of the Canadian Conference on Electrical and Computer Engineering, Vol. 2*, Ontario, Canada, 2-5 May, pp. 677–680.

[12] Sugihara, K. and Smith, J. (1996) A Genetic Algorithm for 3-D Path Planning of a Mobile Robot. Technical report. Tech. Rep. No. 96-09-01. Software Engineering Research Laboratory,University of Hawaii at Manoa.

[13] Bartz-Beielstein, T. and Preuss, M. (2010) Tuning and experimental analysis in evolutionary computation: what we still have wrong. *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, Portland, Oregon, USA GECCO '10, pp. 2625–2646. ACM.

[14] Muñoz, P. and R-Moreno, M. D. (2012) Improving efficiency in any-angle path-planning algorithms. *6th IEEE International Conference on Intelligent Systems IS'12*, Sofia, Bulgaria, 6-8 September.

[15] Muñoz, P., Barrero, D. F., and R-Moreno, M. D. (2013) Statistic methods for path-planning algorithms comparison. *KI - Künstliche Intelligenz*, **27**, 201–211.

[16] Muñoz, P., Barrero, D. F., and R-Moreno, M. D. (2012) Run-time analysis of classical path-planning algorithms. In Bramer, M. and Petridis, M. (eds.), *Research and Development in Intelligent Systems XXIX*, pp. 137–148. Springer London.

[17] Cumming, G. (2012) *Understanding The New Statistics: Effect Sizes, Confidence Intervals, and Meta-Analysis*. Taylor & Francis.

[18] Karaman, S. and Frazzoli, E. (2011) Sampling-based algorithms for optimal motion planning. *I. J. Robotic Res.*, **30**, 846–894.

[19] Janson, L. and Pavone, M. (2013) Fast marching trees: a fast marching sampling-based method for optimal motion planning in many dimensions - extended version. *CoRR*, **abs/1306.3532**.

[20] Dijkstra, E. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**, 269–271.

[21] Botea, A., Muller, M., and Schaeffer, J. (2004) Near optimal hierarchical path-finding. *Journal of Game Development*, **1**, 1–22.

[22] Daniel, K., Nash, A., Koenig, S., and Felner, A. (2010) Theta*: Any-angle path planning on grids. *Journal of Artificial Intelligence Research*, **39**, 533–579.

[23] Ferguson, D. and Stentz, A. (2005) Field D*: An interpolation-based path planner and replanner. *Proceedings of the 12th International Symposium on Robotics Research (ISRR)*, San Francisco, CA, USA, 12–15 October.

[24] Cook, A. F. and Wenk, C. (2011) Link distance and shortest path problems in the plane. *Comput. Geom.*, **44**, 442–455.

[25] Johnson, D. (2002) A theoretician's guide to the experimental analysis of algorithms. *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges Goldwasser, Johnson, and McGeoch, (eds), American Mathematical Society*, **59**, 215–250.

[26] Hooker, J. N. (1995) Testing heuristics: We have it all wrong. *Journal of Heuristics*, **1**, 33–42.

[27] Barr, R., Golden, B., Kelly, J., Resende, M., and Stewart, W. (1995) Designing and Reporting on Computational Experiments with Heuristic Methods. *Journal of Heuristics*, **1**, 9–32.

[28] Kalisiak, M. and van de Panne, M. (2006) Rrt-blossom: Rrt with a local flood-fill behavior. *ICRA*, pp. 1237–1242.

[29] Goldberg, A. V. (1999) Selecting Problems for Algorithm Evaluation. *Proceedings of the 3rd International Workshop on Algorithm Engineering (WAE'99)*, London, UK, pp. 1–11. Springer-Verlag.

[30] Derrac, J., García, S., Molina, D., and Herrera, F. (2011) A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms. *Swarm and Evolutionary Computation*, **1**, 3–18.

[31] Chiarandini, M., Paquete, L., Preuss, M., and Ridge, E. (2007) Experiments on Metaheuristics: Methodological Overview and Open Issues. Technical Report DMF-2007-03-003. The Danish Mathematical Society, Denmark.

[32] Rardin, R. L. and Uzsoy, R. (2001) Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics*, **7**, 261–304.

[33] Birattari, M., Zlochin, M., and Dorigo, M. (2006) Towards a Theory of Practice in Metaheuristics Design: A Machine Learning Perspective. *Theoretical Informatics and Applications*, **40**, 353–369.

[34] Eiben, A. E. and Jelasity, M. 12-17 May A critical note on experimental research methodology in EC. *Proceedings of the 2002 Congress on Evolutionary Computation (CEC2002)*, Honolulu, HI, USA, 12-17 May, pp. 582–587. IEEE.

[35] Hooker, J. N. (1994) Needed: An Empirical Science of Algorithms. *Operations Research*, **42**, 201–212.

[36] Royston, J. (1982) An extension of shapiro and wilk's w test for normality to large samples. *Applied Statistics* , **?**, 115–124.

[37] White, D. R. and Poulding, S. (2009) A Rigorous Evaluation of Crossover and Mutation in Genetic Programming. *Proceedings of the 12th European Conference on Genetic Programming*, Berlin, Heidelberg, April 15-17, pp. 220–231. Springer-Verlag.

[38] Cohen, J. (1988) *Statistical Power Analysis for the Behavioral Sciences, 2nd Ed.*, 2 edition. Routledge.

[39] Barr, R. and Hickman, B. (1993) Reporting Computational Experiments with Parallel Algorithms: Issues, Measures, and Experts' Opinions. *ORSA Journal on Computing*, **5**, 2–18.

[40] Limpert, E., Stahel, W. A., and Abbt, M. (2001) Log-normal Distributions across the Sciences: Keys and Clues. *BioScience*, **51**, 341–352.

[41] Ridge, E. (2007) Design of Experiments for the Tuning of Optimization Algorithms. PhD thesis The University of York. Department of Computer Science.

[42] Bulitko, V., Björnsson, Y., and Lawrence, R. (2010) Case-based subgoaling in real-time heuristic search for video game pathfinding. *Journal of Artificial Intelligence Research*, **39**, 269–300.

[43] Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Thrun, S. (2005) *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.

[44] Brownlee, J. (2007) A Note on Research Methodology and Benchmarking Optimization Algorithms. Technical report. Complex Intelligent Systems Laboratory (CIS), Centre for Information Technology Research (CITR), Faculty of Information and Communication Technologies (ICT), Swinburne University of Technology.

[45] Chiarandini, M., Basso, D., and Stützle, T. (2005) Statistical methods for the comparison of stochastic optimizers. In Et al., K. F. D. (ed.), *The Sixth Metaheuristics International Conference*, Vienna, Austria, 22-26 August, pp. 189–196.

[46] Fleming, P. J. and Wallace, J. J. (1986) How not to Lie with Statistics: The Correct Way to Summarize Benchmark Results. *Communications of the ACM*, **29**, 218–221.

[47] Moret, B. (2002) Towards a Discipline of Experimental Algorithmics. Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges, V.59.

[48] Daida, J., Ross, S., Mcclain, J., Ampy, D., and Holczer, M. (1997) Challenges with Verification, Repeatability, and Meaningful Comparisons in Genetic Programming. *Proceedings of the 2nd Annual Conference on Genetic Programming*, pp. 64–69. Morgan Kaufmann.

[49] Whitley, D., Mathias, K., Rana, S., and Dzubera, J. (1996) Evaluating Evolutionary Algorithms. *Artificial Intelligence*, **85**, 245–276.

[50] Gent, I. P., Grant, S. A., MacIntyre, E., Prosser, P., Shaw, P., Smith, B. M., and Walsh, T. (1997) How Not To Do It. Research report 97.27. School of Computer Studies, University of Leeds.

[51] Erdmann, M. and Lozano-Perez, T. (1987) On multiple moving objects. *Algorithmica*, **2**, 477–521.

[52] Nilsson, N. (1980) Principles of Artificial Intelligence. Tioga Publishing Company, Palo Alto, CA. ISBN 0-935382-01-1.

[53] Epstein, S. and Yun, X. (2010) From Unsolvable to Solvable: An Exploration of Simple Changes. *Workshops at the 24th National Conference on Artificial Intelligence*, Atlanta, Georgia, USA.

[54] Gagliolo, M. and Legrand, C. (2010) Experimental Methods for the Analysis of Optimization Algorithms. Springer-Verlag.

[55] Luke, S. (2001) When Short Runs Beat Long Runs. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, San Francisco, CA, USA, 7-11 July, pp. 74–80.

[56] Barrero, D. F., Castaño, B., R-Moreno, M. D., and Camacho, D. (2011) Statistical Distribution of Generation-to-Success in GP: Application to Model Accumulated Success Probability. *Proceedings of the 14th European Conference on Genetic Programming, (EuroGP 2011)*, Turin, Italy, 27-29 April, LNCS, **6621**, pp. 155–166. Springer Verlag.

[57] Chiarandini, M. and Stützle, T. (2002) Experimental Evaluation of Course Timetabling Algorithms. Technical Report AIDA-02-05. Intellectics Group, Computer Science Department, Darmstadt University of Technology, Darmstadt, Germany.

[58] Frost, D., Rish, I., and Vila, L. (1997) Summarizing CSP Hardness with Continuous Probability Distributions. *Proceedings of the 14th National Conference on Artificial Intelligence and 9th Conference on Innovative Applications of Artificial Intelligence*, Providence, Rhode Island, 14-19 July, pp. 327–333.

[59] Hoos, H. and Stützle, T. (2000) Local Search Algorithms for SAT: An Empirical Evaluation. *Journal of Automated Reasoning*, **24**, 421–481.

[60] Hoos, H. and Stützle, T. (1999) Towards a Characterisation of the Behaviour of Stochastic Local Search Algorithms for SAT. *Artificial Intelligence*, **112**, 213–232.

[61] Hoos, H. and Stützle, T. (1999) Characterizing the Run-Time Behavior of Stochastic Local Search. *Proceedings of the 16th National Conference on Artificial Intelligence*, Orlando, Florida, USA, 18-22 July.

[62] Hoos, H. H. and Stützle, T. (1998) Evaluating Las Vegas Algorithms – Pitfalls and Remedies. *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, WI, USA, 24-26 July, pp. 238–245. Morgan Kaufmann Publishers.

[63] Ribeiro, C. C., Rosseti, I., and Vallejos, R. (2009) On the Use of Run Time Distributions to Evaluate and Compare Stochastic Local Search Algorithms. *Proceedings of the 2nd International Workshop on Engineering Stochastic Local Search Algorithms. Designing, Implementing and Analyzing Effective Heuristics*, Brussels, Belgium, pp. 16–30. Springer-Verlag.

[64] Stützle, T. and Hoos, H. H. (1999) Analyzing the Run-Time Behaviour of Iterated Local Search for the TSP. *Proceedings of the III Metaheuristics International Conference*. Kluwer Academic Publishers.

[65] Ayorkor, G., Stentz, A., and Dias, M. B. 19-23 May Continuous-field path planning with constrained path-dependent state variables. *ICRA 2008 Workshop on Path Planning on Costmaps*, Pasadena, CA, USA, 19-23 May.