Chinese Journal of Aeronautics, (2016), **xxx**(xx): xxx–xxx

Chinese Society of Aeronautics and Astronautics
& Beihang University
**Chinese Journal of Aeronautics**
cja@buaa.edu.cn
www.sciencedirect.com

FULL LENGTH ARTICLE

# Rectangle expansion A* pathfinding for grid maps

**An Zhang** [a,b,*], **Chong Li** [b], **Wenhao Bi** [b]

[a] *School of Aeronautics, Northwestern Polytechnical University, Xian 710129, China*
[b] *School of Electronics and Information, Northwestern Polytechnical University, Xian 710129, China*

Received 25 December 2015; revised 2 March 2016; accepted 22 March 2016

**KEYWORDS**

Breaking path symmetries;
Grid;
Heuristic algorithms;
Path search;
Variant of A*

**Abstract** Search speed, quality of resulting paths and the cost of pre-processing are the principle evaluation metrics of a pathfinding algorithm. In this paper, a new algorithm for grid-based maps, rectangle expansion A* (REA*), is presented that improves the performance of A* significantly. REA* explores maps in units of unblocked rectangles. All unnecessary points inside the rectangles are pruned and boundaries of the rectangles (instead of individual points within those boundaries) are used as search nodes. This makes the algorithm plot fewer points and have a much shorter open list than A*. REA* returns jump and grid-optimal path points, but since the line of sight between jump points is protected by the unblocked rectangles, the resulting path of REA* is usually better than grid-optimal. The algorithm is entirely online and requires no offline pre-processing. Experimental results for typical benchmark problem sets show that REA* can speed up a highly optimized A* by an order of magnitude and more while preserving completeness and optimality. This new algorithm is competitive with other highly successful variants of A*.





## 1. Introduction

Pathfinding is a basic problem in many domains, particularly robotics, artificial intelligence planning, navigation and commercial computer games. A great deal of research has been carried out to improve the quality of resulting paths while keeping costs low. The A* algorithm[1] is now regarded as the gold standard for search algorithms because of its completeness, optimality and effectiveness, and many of the current state-of-the-art pathfinding algorithms are variants of A*.

Search speed, quality of resulting paths and cost of pre-processing are the main evaluation metrics of a pathfinding algorithm. Typical speed enhancements for pathfinding usually involve either trading away optimality for speed or offline pre-processing. The former makes it difficult to guarantee the quality of results. The latter requires extra memory and pre-computing time as seen in hierarchical path-finding A* (HPA*),[2] swamp A*,[3] compressed path databases (CPDs),[4–7] subgoal graphs A* (SUB),[8] and TRANSIT A*.[9–11] Compared with exclusively online algorithms, those employing pre-processing will usually cause poorer performance in non-static environments. On the other hand, algorithms such as theta A*,[12,13] lazy theta A*,[12,14] and A* with post-smoothed paths (A* PS)[15] try to straighten the path during search or as

* Corresponding author at: School of Aeronautics, Northwestern Polytechnical University, Xian 710129, China.
E-mail address: zhangan@nwpu.edu.cn (A. Zhang).
Peer review under responsibility of Editorial Committee of CJA.

ELSEVIER Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.cja.2016.04.023
1000-9361 

part of the post-processing step to obtain better than grid-optimal results and avoid constraining paths to grid edges. This involves expensive visibility tests and sacrifices speed.

In this paper a new optimal algorithm, rectangle expansion A* (REA*), is introduced that explores maps units of unblocked rectangles. Without any pre-processing, REA* can speed up a highly optimized A* by an order of magnitude and more on typical benchmark problem sets. REA* executes purely grid-optimal pathfinding, but the additional benefit of rectangle expansion usually makes the final path better than grid-optimal. This paper makes the following contributions:

(1) A detailed description of the REA* algorithm for 8-connected grid maps.
(2) A theoretical proof for the completeness, optimality of REA* and analysis of its effectiveness.
(3) Trade-off measures to further enhance the method.
(4) Experimental results of REA* and comparisons with standard A* and some of its highly successful variants.

## 2. Related work

The concept of eliminating unnecessary symmetry paths on grid maps has been seen in literature in the last few years, and REA* is inspired by some of the recent successes. The idea of improving a path by reducing unnecessary heading changes also benefits REA*.

REA* is similar to rectangular symmetry reduction (RSR)[16–18] because both divide grid maps into several rectangles free of interior obstacles, and all interior nodes of each rectangle will be pruned during path search. However, they are essentially different. RSR replaces interior nodes with a series of macro-edges, and nodes are visited and expanded individually during search. In REA*, each rectangle is operated in as a whole, ignoring all unnecessary interior connections. Outer perimeter nodes, continuously free of obstacles, will be considered “search nodes”, which means fewer and quicker list operations. What’s more, REA* is carried out entirely online while RSR requires offline pre-processing to divide grid maps into rectangles and assign nodes.

Block A*[19] is another algorithm that searches using rectangles. Information about all possible distances across a block of grid cells ($m \times n$ region of nodes) is pre-computed and stored in a new type of database, the local distance database (LDDB). Distances between boundary points are queried from the LDDB during an online A* search. The number of entries in the LDDB increases extremely fast, so the size of blocks cannot be large (no more than $5 \times 5$ in Ref. 19). Block A* requires to pre-compute the LDDB and, though different LDDBs can be chosen to improve the resulting path, it is only guaranteed to be optimal in 4-connected grid maps.

Jump point search (JPS)[20,21] is the current state-of-the-art online algorithm. JPS identifies and selectively expands only certain nodes in a grid map called jump points. All intermediate nodes on paths connecting two jump points are never expanded. The algorithm can speed up A* greatly and, just like REA*, returns skipped path points. However, while JPS guarantees grid-optimal paths, REA* returns paths better than grid-optimal.

Anya A*[22] is a recently proposed algorithm for grid maps that uses contiguous sets of nodes in horizontal or vertical lines as search nodes. Anya A* expands search nodes from each line to a neighboring line with frequent, expensive line-of-sight tests while REA* expands search nodes in units of rectangles, a much cheaper operation. But, Anya A* offers any-angle optimal pathfinding that is not artificially constrained to the points of a grid.

## 3. Rectangle expansion A* algorithm

A grid map is one of the most popular types of maps used to represent realistic terrain in literature. Assuming that (in 8-connected grid maps) an agent operates on a grid map with obstacles consisting of blocked cells and traversable areas consisting of unblocked cells, it can move from any unblocked grid center to another cardinally or diagonally if both adjacent cardinal directions are unblocked.

REA* is a variant of standard A*. Pseudo-code of REA* is shown in Table 1. The octile distance is used to estimate the distance between two cells heuristically. The respective lengths of cardinal and diagonal moves are 1 and 1.414. A matrix the same size as the map is used to store all the grid points. Point $(x, y)$ represents the point at the intersection of $x$th column and $y$th row in the grid map with (1, 1) as the point in the upper left corner. The octile distance between $p(x, y)$ and $p'(x', y')$ is:

$$\mathrm{octile}(p, p') = 1.414 \times \min(\Delta x, \Delta y) + |\Delta x - \Delta y| \quad (1)$$

where $\Delta x = |x - x'|$ and $\Delta y = |y - y'|$.

**Definition 1.** A grid interval $I$ is a set of contiguous points in the same row or column of the grid. If all points in $I$ are unblocked, $I$ is an unblocked interval. Each interval can be defined in terms of its endpoints $a$ and $b$, written as $[a, b]$.

Search nodes in REA* are not individual cells in the grid map but unblocked intervals of the map. To distinguish them from our search nodes, traditional individual grid cells will be called cells or points in this paper for simplicity. The key idea of REA* is that, when exploring a grid map, REA* doesn’t visit individual cells one by one. Instead, a linear search node will expand an unblocked rectangle until stopped by obstacle cells. Only interior boundary cells of the expanded unblocked rectangle will be visited by parent cells from the original search node, and unblocked points inside the rectangle will be pruned. Then, interior boundaries of the rectangle (except the original

**Table 1** REA*() algorithm.

Require: $S$: the start point, $G$: the goal point;

```
Initialize();
if InsertS() then
    return “path is found”;
While (Openlist! = ∅)
    CBN: = the current best search node;
    if Expand (CBN) then
        return “path is found”;
return “no path is found”;
```