R* Search: The Proofs

Maxim Likhachev
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
maximl@seas.upenn.edu

Anthony Stentz
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
axs@rec.ri.cmu.edu

1 Introduction

The pseudocode in Figure 1 is slightly different from the one presented in the main paper. In particular, every state s now maintains an additional variable, $v(s)$, which is initially set to ∞, and then is reset to the g-value of s every time s is expanded. This modification simplifies the proofs. Otherwise, the v-values are not used in the algorithm, and therefore it should be clear that the pseudocode in Figure 1 is algorithmically identical to the pseudocode of R* as presented in the main text of the paper.

Henceforth, all line numbers in the text of the proofs will refer to the pseudocode in Figure 1.

2 Notations and Assumptions

- $c(s, s') > 0$ - the cost of a transition between states s and $s' \in succ(s)$ in the original graph
- ϵ - the inflation factor. We restrict that $1 \leq \epsilon < \infty$.
- $\pi(u, v)$ - a path from state u to state v in the original graph
- $c(\pi(u, v))$ - the actual cost of path $\pi(u, v)$ which is the summation of the costs of the transitions on the path

- $\pi_{opt}(u,v) = \arg\min_{\pi(u,v)} c(\pi(u,v))$ - an optimal path from u to v in the original graph
- $c^*(u,v) = c(\pi_{opt}(u,v))$ - the cost of an optimal path from u to v in the original graph
- $h(s,s')$ - heuristics. It (under) estimates the cost of an optimal path from s to s'. It needs to be consistent: $h(s, s_{\text{goal}}) \leq c(s,s') + h(s', s_{\text{goal}})$ for any s and any successor s' of s if $s \neq s_{\text{goal}}$ and $h(s, s_{\text{goal}}) = 0$ if $s = s_{\text{goal}}$.
- Γ - the graph of all states generated by R* plus s_{goal} if it hasn't been generated yet. Thus, initially Γ contains only s_{start} and s_{goal}. Afterwards, every time any state s is expanded, its successors - the states that appear in the set $SUCCS(s)$ on line 26 - and the edges from s to these states are added to Γ
- $c(path_{s,s'})$ - the actual cost of the path encoded in $path_{s,s'}$. It is assumed to be infinite if $path_{s,s'} = \textbf{null}$.
- $\pi^\Gamma(u,v)$ - a path from $u \in \Gamma$ to $v \in \Gamma$ in graph Γ using the edges in Γ
- $\pi^\Gamma_{bp}(u,v)$ - a path $\pi^\Gamma(u,v)$ re-constructed using backpointers. In other words, any two consecutive states $s_i \in \Gamma$ and $s_{i+1} \in \Gamma$ on the path are such that $s_i = bp(s_{i+1})$.
- $c_{low}(\pi^\Gamma(u,v))$ - the cost of the path $\pi^\Gamma(u,v)$ in which the cost of a transition between any two consecutive states $s_i \in \Gamma$ and $s_{i+1} \in \Gamma$ on the path is equal to the variable $c_{low}(path_{s_i,s_{i+1}})$. It is assumed to be infinite, if $\pi^\Gamma_{bp}(u,v)$ contains a state $s \neq u$ such that $bp(s) = \textbf{null}$.
- $\pi^\Gamma_{opt}(u,v) = \arg\min_{\pi^\Gamma(u,v)} c_{low}(\pi^\Gamma(u,v))$ - an optimal path in between states u and v in terms of costs c_{low}
- $c(\pi^\Gamma(u,v))$ - the actual cost of the path $\pi^\Gamma(u,v)$ in which the cost of a transition between any two consecutive states $s_i \in \Gamma$ and $s_{i+1} \in \Gamma$ on the path is equal to $c(path_{s_i,s_{i+1}})$ - the actual cost of the path $path_{s_i,s_{i+1}}$. It is assumed to be infinite, if $\pi^\Gamma_{bp}(u,v)$ contains a state $s \neq u$ such that $bp(s) = \textbf{null}$.

- $c^*(\pi^\Gamma(u,v))$ - the cost of the path $\pi^\Gamma(u,v)$ in which the cost of a transition between any two consecutive states $s_i \in \Gamma$ and $s_{i+1} \in \Gamma$ on the path is equal to $c^*(path_{s_i,s_{i+1}})$ - the cost of an optimal path from s_i to s_{i+1} in the original graph. It is assumed to be infinite, if $\pi^\Gamma_{bp}(u,v)$ contains a state $s \neq u$ such that $bp(s) = \mathbf{null}$.

Let us also define $g_\epsilon(s) = \min_{s'|s \in SUCC(s')}(v(s') + \epsilon\, c(s',s))$ if $s \neq s_{start}$ and $g_\epsilon(s) = 0$ otherwise. We also assume that min operation over an empty set returns a vector of the expected dimensions in which each of its dimensions is set to ∞. For example, $\min_{s \in OPEN}(\mathrm{k}(s)) = [\infty;\infty]$ if $OPEN = \emptyset$.

Finally, we also assume the following about the TryToComputeLocalPath function.

Assumption 1 *Right after the execution of line 7 the following holds:*

- $c_{low}(path_{bp(s),s}) \leq \epsilon\, c^*(bp(s),s)$ *if* $path_{bp(s),s} = \mathbf{null}$*;*
- $c(path_{bp(s),s}) \leq c_{low}(path_{bp(s),s}) \leq \epsilon\, c^*(bp(s),s)$ *if* $path_{bp(s),s} \neq \mathbf{null}$*;*

3 Basic Theorems

Lemma 1 *At line 17, the following holds for all $s, s' \in \Gamma$ such that $s' \in SUCCS(s)$:*

- $c_{low}(path_{s,s'}) \leq \epsilon\, c^*(s,s')$ *if* $path_{s,s'} = \mathbf{null}$*;*
- $c(path_{s,s'}) \leq c_{low}(path_{s,s'}) \leq \epsilon\, c^*(s,s')$ *if* $path_{s,s'} \neq \mathbf{null}$*;*

Proof:

For each state s, the set $SUCCS(s)$ is created on lines 23-26. Right after it, for each member s' of $SUCCS(s)$, we set $path_{s,s'} = \mathbf{null}$ and $c_{low}(path_{s,s'}) = h(s,s') \leq c^*(s,s')$ which is consistent with the theorem.

Afterwards, the only place where we can potentially modify either $path(s,s')$ or $c_{low}(path_{s,s'})$ is on line 7. However, the modifications are consistent with the statement of the theorem according to the assumption 1. ■

1 **procedure UpdateState**(s)
2 if ($g(s) > \epsilon\, h(s_{\text{start}}, s)$ OR
($path_{bp(s),s} = \textbf{null}$ AND path to s seems to be too hard to compute))
3 insert/update s in *OPEN* with priority $k(s) = [1,\ g(s) + \epsilon\, h(s, s_{\text{goal}})]$;
4 else
5 insert/update s in *OPEN* with priority $k(s) = [0,\ g(s) + \epsilon\, h(s, s_{\text{goal}})]$;

6 **procedure ReevaluateState**(s)
7 $[path_{bp(s),s}, c_{low}(path_{bp(s),s})]$ = TrytoComputeLocalPath($bp(s)$, s);
8 if ($path_{bp(s),s} = \textbf{null}$ OR $g(bp(s)) + c_{low}(path_{bp(s),s}) > \epsilon\, h(s_{\text{start}}, s)$)
9 $bp(s) = \arg\min_{s' | s \in SUCCS(s')}(g(s') + c_{low}(path_{s',s}))$;
10 $g(s) = g(bp(s)) + c_{low}(path_{bp(s),s})$;
11 UpdateState(s);

12 **procedure PESS**()
13 $g(s_{\text{goal}}) = v(s_{\text{goal}}) = \infty$, $bp(s_{\text{goal}}) = bp(s_{\text{start}}) = \textbf{null}$, $k(s_{\text{goal}}) = [1, \infty]$;
14 $OPEN = CLOSED = \emptyset$;
15 $g(s_{\text{start}}) = 0$, $v(s_{\text{start}}) = \infty$;
16 insert s_{start} into *OPEN* with priority $k(s_{\text{start}}) = [0, \epsilon\, h(s_{\text{start}}, s_{\text{goal}})]$;
17 while ($k(s_{\text{goal}}) \geq \min_{s' \in OPEN} k(s')$ AND $OPEN \neq \emptyset$)
18 remove s with the smallest priority from *OPEN*;
19 if $s \neq s_{\text{start}}$ AND $path_{bp(s),s} = \textbf{null}$
20 ReevaluateState(s);
21 else
22 $v(s) = g(s)$; insert s into *CLOSED*;
23 let *SUCCS* be the set of K randomly chosen states at distance Δ from s
24 if distance from s_{goal} to s is smaller than or equal to Δ
25 $SUCCS(s) = SUCCS(s) \cup \{s_{\text{goal}}\}$;
26 $SUCCS(s) = SUCCS(s) - SUCCS(s) \cap CLOSED$
27 for each state $s' \in SUCCS(s)$
28 $[path_{s,s'}, c_{low}(path_{s,s'})] = [\textbf{null}, h(s, s')]$;
29 if s' is visited for the first time
30 $g(s') = v(s') = \infty$, $bp(s') = \textbf{null}$;
31 if $bp(s') = \textbf{null}$ OR $g(s) + c_{low}(path_{s,s'}) < g(s')$
32 $g(s') = g(s) + c_{low}(path_{s,s'})$; $bp(s') = s$;
33 UpdateState(s');

Figure 1: The pseudocode of R*

Lemma 2 *At line 17, $OPEN \cup CLOSED$ contains all and only states in Γ except possibly for states with infinite g-values. Also, $OPEN \cap CLOSED = \emptyset$.*

Proof: We prove by induction. By definition, Γ contains s_{start}, s_{goal}, and all states generated on line 23. During the first execution of line 17, no states other than s_{start} and s_{goal} have been generated, s_{start} in *OPEN* and $g(s_{\text{goal}}) = \infty$. Thus, the theorem holds at this point.

Now suppose the theorem holds at the ith execution of line 17. We need to show that it continues to hold during the next execution of line 17. First, a state s is removed from $OPEN$ on line 18, but then it is either reevaluated on line 20, in which case it is inserted into $OPEN$ at the end of the function, or it is expanded and therefore is inserted into $CLOSED$ on line 22. Thus, s remains in either $OPEN$ or $CLOSED$ but not both.

The set of states in Γ is then grown on line 23. On line 26 we trim $SUCCS(s)$ to contain only the states that are not in $CLOSED$. By induction then, if a state $s' \in SUCCS(s)$ then it is either in $OPEN$, or it has infinite g-value, or it has never been generated before. In the latter case, its g-value is set to ∞ right afterwards (line 30). Thus, now every state $s' \in SUCCS(s)$ is either in $OPEN$ or has infinite g-value. The g-values of states in $SUCCS(s)$ can potentially decrease (and become finite) on line 32. But then every state whose g-value decreases is inserted into $OPEN$ during the call to the UpdateState function on on the next line. Thus, the theorem continues to hold during the next execution of line 17 and holds during every execution of this line by induction. ■

Lemma 3 *Suppose state s is being expanded on line 22. Then the next time line 17 is executed $v(s) = g(s)$, where $g(s)$ before and after the expansion of s is the same. Afterwards, $v(s)$ and $g(s)$ remain the same until the algorithm terminates.*

Proof: $v(s)$ is set to $g(s)$ right on line 22. We thus only need to show that $g(s)$ does not change while s is being expanded. This follows from the fact that at the time s is selected for expansion it is added to $CLOSED$. Consequently, s can not be in the set $SUCCS(s)$ since all states in $CLOSED$ are removed from $SUCCS(s)$ on line 26. As a result, $g(s)$ can not be changed since the only states whose g-values get modified are those that are in the set $SUCCS(s)$.

Once state s gets inserted into $CLOSED$ it is never removed from it. Thus, from lemma 2 it follows that it never gets inserted into $OPEN$ and therefore can not be re-expanded. As a result, its v-value can not be modified since it is only modified for states that are either expanded (line 22) or states that have not been visited before (line 30). $g(s)$ can not be modified either since it is modified only for states selected from $OPEN$ for reevaluation (line 20) and for states that are successors of the state that is being expanded (lines 30 and 32) and the set $SUCCS$ does not contain any states that are in *CLOSED* due to line 26. Thus, $v(s)$ and $g(s)$ remain the same until the algorithm terminates. ■

Lemma 4 *On line 17, for any state $s \in \Gamma$, $v(s) \geq g(s)$.*

Proof: We prove by induction. The theorem clearly holds during the first execution of line 17 since $g(s_{\text{start}}) = 0 < v(s_{\text{start}}) = \infty$, $g(s_{\text{goal}}) = v(s_{\text{goal}}) = \infty$ and there are no other states in Γ at this point.

Now suppose the theorem holds at the ith execution of line 17. We need to show that it continues to hold during the next execution of line 17. All the states added to Γ have their v-values set to ∞ on line 30. The only place where a v-value is changed from infinity to a potentially finite value is line 22. It is set there to the g-value of the state. However, on the same line, the state is inserted into *CLOSED* and its g-value never changes afterwards according to lemma 3. Thus, the theorem continues to hold during the next execution of line 17 and holds during every execution of this line by induction. ■

Lemma 5 *No state $s \in \Gamma$ is selected more than once for expansion on line 22.*

Proof: Once a state s is selected for expansion on line 22, it is inserted into *CLOSED*. Since it never gets removed from *CLOSED*, it will never reappear in *OPEN* according to lemma 2 and will therefore never again will be selected for re-expansion. ■

Lemma 6 *At line 17, for any state $s \in$ OPEN it holds that if the first element of $k(s)$ is 0, then $g(s) \leq \epsilon\, h(s_{\text{start}}, s)$.*

Proof: The statement holds during the first execution of line 17 because *OPEN* contains only s_{start} whose g-value is 0.

Afterwards, every time a g-value of any state s is modified (lines 10 and 32) or initialized (line 30), the function UpdateState is called on it which updates $k(s)$ to be consistent with the statement of the theorem. ■

Theorem 1 *At line 17, $g(s_{\text{start}}) = 0$, and for $\forall s \in \Gamma$ such that $s \neq s_{\text{start}}$, the following inequality holds*

- $g(s) = v(bp(s)) + c_{low}(path_{bp(s),s})$ *if* $bp(s) \neq$ **null***;*

- $g(s) = \infty$ *if* $bp(s) =$ **null**;

and at least one of the following inequalities holds:

- $g(s) \leq \epsilon\, h(s_{\text{start}}, s)$
- $g(s) = \min_{s'|s \in SUCCS(s')}(v(s') + c_{low}(path_{s',s}))$

Proof:

Let us first prove the first part of the theorem: $g(s_{\text{start}}) = 0$ and for every other state $s \in \Gamma$, $g(s) = v(bp(s)) + c_{low}(path_{bp(s),s})$ if $bp(s) \neq$ **null** and $g(s) = \infty$ otherwise. We prove by induction. This holds during the first execution of line 17 because $g(s_{\text{start}}) = 0$, $g(s_{\text{goal}}) = \infty$, $bp(s_{\text{goal}}) =$ **null** and there are no other states in Γ.

Afterwards, every time bp-value is set to **null** for newly generated states on line 30, the g-value of the state is also set to ∞ on the same line. A g-value, v-value, bp-value and $c_{low}(path_{s',s})$ may also change on line 22, lines 28-32 and during the call of the ReevaluateState function on line 20. In the first case, we expand state s for the first time according to lemma 5 and therefore there are no successors of s yet and the change to its v-value does not affect any other states. Once the successors of s are generated on lines 23-26, the execution of line 32 makes sure that $g(s') = g(bp(s')) + c_{low}(path_{bp(s'),s'})$ if $bp(s') = s$. Moreover, according to lemma 3, $g(s) = v(s)$. Thus, $g(s') = v(bp(s')) + c_{low}(path_{bp(s'),s'})$. s' can not be s_{start} because s_{start} gets expanded and inserted into $OPEN$ at the beginning of the first iteration of the while loop and therefore subsequent $SUCCS(s)$ sets can not contain s_{start} because of line 26. Therefore, $g(s_{\text{start}})$ continues to be zero. Thus, the theorem continues to hold after a state is expanded.

In case of calling the ReevaluateState function on state s on line 20, the execution of line 10 makes again certain that $g(s) = v(bp(s)) + c_{low}(path_{bp(s),s})$, because in order for $s \in SUCCS(bp(s))$, $bp(s)$ must have been expanded and therefore $g(bp(s)) = v(bp(s))$ according to lemma 3. Also, $bp(s) \neq$ **null** because bp-values are set to non-**null** values before each call to UpdateState function which inserts states into $OPEN$. It also remains that $g(s_{\text{start}}) = 0$ since the the ReevaluateState function could not have been called on s_{start} because of the test on line 19. Thus, the first part of the theorem continues to hold during the next execution of line 17 and holds during every execution of this line by induction.

We now prove the second part of the theorem, namely that for every state $s \in \Gamma$, $s \neq s_{\text{start}}$, either $g(s) \leq \epsilon\, h(s_{\text{start}}, s)$ or $g(s) = \min_{s' | s \in SUCCS(s')}(v(s') + c_{low}(path_{s',s}))$, or both. We again prove by induction. It holds during the first execution of line 17 because apart from s_{start}, Γ contains only s_{goal} and $g(s_{\text{goal}}) = \infty = \min_{s' | s_{\text{goal}} \in SUCCS(s')}(v(s') + c_{low}(path_{s',s_{\text{goal}}}))$.

Now suppose the second part of the theorem holds at the ith execution of line 17. We need to show that it continues to hold during the next execution of line 17.

This part of the theorem can be affected by changes in v-values, g-values and c_{low}-values. First, let us consider the changes in v-values. In the main body of the while loop, there are two places where v-values change: line 22 and line 30. In the latter case, v is set to ∞ for states that have never been visited and therefore could not have had any successors. This operation therefore can not break the theorem. In the former case, $v(s)$ is set to $g(s)$. However, for every successor s' of s, its g-value is decreased so that it is consistent with the statement of the theorem on line 32 since $g(s) = v(s)$ at this point because $v(s)$ was set to $g(s)$ on line 22 and $g(s)$ remained the same according to lemma 3.

Now let us consider the changes in g- and c_{low}-values. They happen on lines 28, 30, 32, 7 and 10. In case $c_{low}(path_{s,s'})$ is set on line 28, then $g(s')$ is updated appropriately on lines 31-32, where $g(s) = v(s)$ as we have just shown. In case the g-value of a newly generated state is set to ∞ on line 30, then it does not yet have any predecessors.

Now consider the changes on lines 7 and 10. Before entering the ReevaluateState function, the theorem holds by induction. Suppose the test on line 8 succeeds. Then the execution of line 10 results in $g(s) = \min_{s' | s \in SUCCS(s')}(g(s') + c_{low}(path_{s',s}))$ which is consistent with the theorem since $g(s') = v(s')$ because s' must have been in *CLOSED* in order for it to have s as a successor and all states in *CLOSED* have their g-values equal to their v-values according to lemma 3.

Now suppose the test on line 8 fails. This means that $g(bp(s)) + c_{low}(path_{bp(s),s}) \leq \epsilon\, h(s_{\text{start}}, s))$ and therefore the execution of line 10 results in $g(s) \leq \epsilon\, h(s_{\text{start}}, s))$. Thus, the second part of the theorem also continues to hold during the next execution of line 17 and holds during every execution of this line by induction. ■

Theorem 2 *At line 17, for any state* $s \in \Gamma$, $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s)) \leq g(s) \leq v(s)$.

Proof: $v(s) \geq g(s)$ holds according to Lemma 4. We thus only need to show that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s)) \leq g(s)$. The statement follows if $g(s) = \infty$. We thus assume a finite g-value.

Consider a path $\pi^{\Gamma}_{bp}(s_{\text{start}}, s)$ from s_{start} to s: $s_0 = s_{start}, s_1, ..., s_k = s$, where $s_i \in \Gamma$ for all $0 \leq i \leq k$. Then from the definition of such path for any $i > 0$, $g(s_i) = v(s_{i-1}) + c_{low}(path_{s_{i-1},s_i}) \geq g(s_{i-1}) + c_{low}(path_{s_{i-1},s_i})$ from theorem 1 and lemma 4. For $i = 0$, $g(s_i) = g(s_{start}) = 0$. Thus, $g(s) = g(s_k) \geq g(s_{k-1}) + c_{low}(path_{s_{k-1},s_k}) \geq g(s_{k-2}) + c_{low}(path_{s_{k-2},s_{k-1}}) + c_{low}(path_{s_{k-1},s_k}) \geq ... \geq \sum_{j=1..k} c_{low}(path_{s_{j-1},s_j})$. Moreover, from lemma 1 it follows that $g(s) \geq \sum_{j=1..k} c(path_{s_{j-1},s_j}) = c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s))$. ■

4 Main Theorems

For the purpose of the following few theorems we will introduce the following set Q:

$$Q = \{u \mid v(u) > g_{\epsilon}(u) \wedge v(u) > \epsilon * c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, u))\} \tag{1}$$

The set Q takes the place of the *OPEN* list in the next theorem. In particular, Theorem 3 says that all states s in Γ which are ahead of Q have their g-values within a factor of ϵ of $c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$. Theorem 4 builds on this result by showing that *OPEN* is always a superset of Q, and therefore the states which are ahead of *OPEN* are also ahead of Q.

Theorem 3 *At line 17, let Q be defined according to the definition 1. Then for any state $s \in \Gamma$ with $(g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})\ \forall u \in Q)$, it holds that $g(s) \leq \epsilon\, c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$.*

Proof: We prove by contradiction. Suppose there exists an s such that $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})\ \forall u \in Q$, but $g(s) > \epsilon\, c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$. The latter implies that $c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s)) < \infty$. We also assume that $s \neq s_{start}$ since otherwise $g(s) = 0 = \epsilon\, c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$ from Theorem 1.

Consider a path $\pi^{\Gamma}_{opt}(s_{\text{start}}, s)$ from s_{start} to s, $\pi(s_0 = s_{start}, ..., s_k = s)$. Such path must exist since $c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s)) < \infty$. Our assumption that $g(s) > \epsilon\, c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s)) \geq \epsilon\, h(s_{\text{start}}, s)$ means that there exists at least one $s_i \in \pi(s_0, ..., s_{k-1})$ whose $v(s_i) > \epsilon\, c^{*}(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i))$. Otherwise,

$$g(s) = g(s_k) = \min_{s' \mid s \in SUCCS(s')} (v(s') + c_{low}(path_{s',s_k})) \quad \leq \quad //theorem\ 1$$

$$
\begin{aligned}
v(s_{k-1}) + c_{low}(path_{s_{k-1},s_k}) &\leq \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{k-1})) + c_{low}(path_{s_{k-1},s_k}) &\leq \\
\epsilon\,(c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{k-1})) + c^*(s_{k-1}, s_k)) &= \quad //lemma\ 1 \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_k)) &= \quad \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))
\end{aligned}
$$

Let us now consider $s_i \in \pi(s_0, ..., s_{k-1})$ with the smallest index $i \geq 0$ (that is, the closest to s_{start}) such that $v(s_i) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i))$. We will now show that $s_i \in Q$. If $i = 0$ then $g_\epsilon(s_i) = g_\epsilon(s_{start}) = 0$ according to the definition of the g_ϵ-values. Thus: $v(s_i) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i)) = 0 = g_\epsilon(s_i)$, and $s_i \in Q$. If $i > 0$ then

$$
\begin{aligned}
v(s_i) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i)) &= \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{i-1})) + \epsilon\, c^*(s_{i-1}, s_i) &\geq \\
v(s_{i-1}) + \epsilon\, c^*(s_{i-1}, s_i) &
\end{aligned}
$$

since we picked s_i to be the closest state to s_{start} with $v(s_i) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i))$. Thus,

$$
\begin{aligned}
v(s_i) > v(s_{i-1}) + \epsilon\, c^*(s_{i-1}, s_i) &\geq \\
\min_{s' | s_i \in SUCCS(s')} (v(s') + \epsilon\, c^*(s', s_i)) &= \quad g_\epsilon(s_i)
\end{aligned}
$$

As such, it must again be the case that $s_i \in Q$.

We will now also show that $g(s_i) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i))$. It is clearly so when $i = 0$ according to Theorem 1. For $i > 0$, if $g(s_i) \leq \epsilon\, h(s_{\text{start}}, s_i)$ then $g(s_i) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i))$ due to the admissibility of heuristics and otherwise, from theorem 1, it follows that

$$
\begin{aligned}
g(s_i) \leq \min_{s' | s_i \in SUCCS(s')} (v(s') + c_{low}(path_{s',s_i})) &\leq \\
v(s_{i-1}) + c_{low}(path_{s_{i-1},s_i}) &\leq \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{i-1})) + c_{low}(path_{s_{i-1},s_i}) &\leq \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{i-1})) + \epsilon\, c^*(s_{i-1}, s_i) &\leq \\
\epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i)) &
\end{aligned}
$$

We will now show that $g(s) + \epsilon\, h(s, s_{\text{goal}}) > g(s_i) + \epsilon\, h(s_i, s_{\text{goal}})$, and finally arrive at a contradiction. According to our assumption

$$
\begin{aligned}
g(s) > \epsilon\, c^*(\pi^\Gamma_{opt}(s_{\text{start}}, s)) &= \\
\epsilon\, c^*(\pi^\Gamma_{opt}(s_0, s_i)) + \epsilon\, c^*(\pi^\Gamma_{opt}(s_i, s_k)) &\geq \\
g(s_i) + \epsilon\, c^*(\pi^\Gamma_{opt}(s_i, s_k))
\end{aligned}
$$

Adding $\epsilon\, h(s, s_{\text{goal}})$ on both sides and using the consistency of heuristics:

$$
\begin{aligned}
g(s) + \epsilon\, h(s, s_{\text{goal}}) &> \\
g(s_i) + \epsilon\, c^*(\pi^\Gamma_{opt}(s_i, s)) + \epsilon\, h(s, s_{\text{goal}}) &\geq \\
g(s_i) + \epsilon\, (c^*(s_i, s) + h(s, s_{\text{goal}})) &\geq \\
g(s_i) + \epsilon\, h(s_i, s_{\text{goal}})
\end{aligned}
$$

The inequality $g(s) + \epsilon\, h(s, s_{\text{goal}}) > g(s_i) + \epsilon\, h(s_i, s_{\text{goal}})$ implies, however, that $s_i \notin Q$ since $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})\ \forall u \in Q$. But this contradicts to what we have proven earlier. ■

Theorem 4 *At line 17, for any state $s \in \Gamma$ with $(g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})\ \forall u \in OPEN)$, it holds that $g(s) \leq \epsilon\, c^*(\pi^\Gamma_{opt}(s_{\text{start}}, s))$.*

Proof: Let Q be defined according to the definition 1. Consider the very first time line 17 gets executed. At this point, *OPEN* contains only s_{start} for which $g(s_{\text{start}}) = 0 \leq \epsilon\, c^*(\pi^\Gamma_{opt}(s_{\text{start}}, s_{\text{start}})) = 0$. All other states in Γ, namely, s_{goal} have infinite g-values. Thus, the theorem holds. Also, during the first execution of line 17, $CLOSED = \emptyset$, and therefore, the following statement, denoted by (*), holds: for any state $s \in CLOSED$ $v(s) \leq \epsilon\, c^*(\pi^\Gamma_{opt}(s_{\text{start}}, s))$.

We will now show by induction that the theorem continues to hold for the consecutive executions of the line 17 *within* each call to the ImprovePath function. Suppose the theorem and the statement (*) held during all the previous executions of line 17, and they still hold when a state s is selected on line 18. We need to show that the theorem holds the next time line 17 is executed.

We first prove that the statement (*) still holds during the next execution of line 17. If test on line 19 succeeds and s does not get expanded, then nothing is added to *CLOSED* and therefore statement (*) continues to hold. Otherwise, since the v-value of only s is being changed and only s is being added to *CLOSED*, we only need to show that $v(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$ during the next execution of line 17 (that is, after the expansion of s). If the first element of $k(s)$ is equal to 0, then according to lemma 6, $g(s) \leq \epsilon\, h(s_{\text{start}}, s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$. If the first element of $k(s)$ is 1, then since when s is selected on line 18 $k(s) = \min_{u \in OPEN}(k(u))$, we have $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})$ $\forall u \in OPEN$. According to the assumptions of our induction then $g(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$. As a result, from lemma 3 it then follows that the next time line 17 is executed $v(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$, and hence the statement (*) still holds.

We now prove that the theorem itself also holds during the next execution of line 17. We prove it by showing that Q continues to be a subset of *OPEN* the next time line 17 is executed. According to lemma 2, *OPEN* set contains all states in Γ that are not in *CLOSED* except possibly for s_{goal} if its g-value is infinite. Since, as we have just proved, the statement (*) holds the next time line 17 is executed, all states s in *CLOSED* set have $v(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$. Thus, any state s that has $v(s) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$, except possibly for s_{goal} if its g-value is infinite, is guaranteed to be in *OPEN*. Now consider any state $u \in Q$. Then $v(u) > \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, u))$ according to the definition of Q. Thus, $u \in OPEN$ unless $u = s_{\text{goal}}$ with $g(u) = \infty$. However, if $g(u) = \infty$, then according to lemma 4, $v(u) = g(u) = \infty$ and according to theorem 1, either $v(u) = g(u) \leq \epsilon\, h(s_{\text{start}}, u) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, u))$ or $v(u) = g(u) = \min_{s'|u \in SUCCS(s')}(v(s') + c_{low}(path_{s',u})) \leq \min_{s'|u \in SUCCS(s')}(v(s') + \epsilon\, c(path_{s',u})) = g_{\epsilon}(u)$. Therefore, if $g(u) = \infty$, then $u \notin Q$. This shows that $Q \subseteq OPEN$.

Consequently, if any state s has $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})$ $\forall u \in OPEN$, it is also true that $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(u) + \epsilon\, h(u, s_{\text{goal}})$ $\forall u \in Q$, and $g(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$ from Theorem 3. This proves that the theorem holds during the next execution of line 17, and proves the whole theorem by induction. ∎

Theorem 5 *At line 17, for any state* $s \in \text{CLOSED}$, $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s)) \leq g(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$.

Proof: The fact that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s)) \leq g(s)$ follows from theorem 2. To prove that $g(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$ if $s \in CLOSED$, consider s at the time it is being expanded (line 22). We distinguish two different scenarios. First, suppose the first element of $k(s)$ is 0. Then according to lemma 6, $g(s) \leq \epsilon\, h(s_{\text{start}}, s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$.

Now suppose the first element of $k(s)$ is 1. Because $k(s) = \arg\min_{s' \in OPEN} k(s')$ since this is how s was selected on line 18, it follows that any other state s' in *OPEN* has $k(s') \geq k(s)$. This implies that the first element of $k(s')$ is also 1 and the second element of $k(s')$ is at least as large as $k(s)$. That is, $g(s) + \epsilon\, h(s, s_{\text{goal}}) \leq g(s') + \epsilon\, h(s', s_{\text{goal}})$ for every state $s' \in OPEN$. From theorem 4 it then follows that $g(s) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s))$.

Since the g-values and v-values of states in *CLOSED* do not change according to lemma 3, the theorem holds. ■

Theorem 6 *Suppose on line 23 R* always generates all of the states that lie at distance Δ from s. Then upon termination, R* returns a path whose cost is no more $g(s_{\text{goal}})$ which, in turn, is no more than ϵ times the cost of an optimal path from s_{start} to s_{goal}. That is, $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq g(s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$.*

Proof: The fact that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq g(s_{\text{goal}})$ follows from theorem 2. To prove that $g(s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ we first show that $g(s_{\text{goal}}) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{\text{goal}}))$ and then show that either $g(s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ or Γ must contain a path $\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})$ such that $c^*(\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})) = c^*(s_{\text{start}}, s_{\text{goal}})$, which also implies the desired inequality.

We distinguish two scenarios. Suppose s_{goal} has never been inserted into *OPEN* during the execution of R*. Then its g-value remains to be equal to ∞ and its priority remains to be equal to $[1; \infty]$ as they were set initially. According to the termination condition of the while loop, it holds that either $OPEN = \emptyset$ or $\min_{s' \in OPEN} k(s') > k(s_{\text{goal}})$. In the former case, $\min_{s' \in OPEN} g(s') + \epsilon\, h(s', s_{\text{goal}}) = \infty = g(s_{\text{goal}}) + \epsilon\, h(s_{\text{goal}}, s_{\text{goal}})$ using our convention that min operator over an empty set returns infinity. In the latter case, the first elements of the priorities of the states in *OPEN* are 1s and the second elements are infinite, implying again that $\min_{s' \in OPEN} g(s') + \epsilon\, h(s', s_{\text{goal}}) = \infty = g(s_{\text{goal}}) + \epsilon\, h(s_{\text{goal}}, s_{\text{goal}})$. Thus, from theorem 4, it follows that $g(s_{\text{goal}}) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{\text{goal}}))$.

Now suppose s_{goal} has been inserted into *OPEN* during the execution of R*. Then the only way for it to have been removed from *OPEN* without being reinserted immediately afterwards is by being expanded. In this case, it must have been added to *CLOSED* on line 22. The fact that $g(s_{\text{goal}}) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{\text{goal}}))$ then follows from theorem 5.

We now prove that either $g(s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ or Γ must contain a path $\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})$ such that $c^*(\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})) = c^*(s_{\text{start}}, s_{\text{goal}})$. We prove by contradiction and assume that $g(s_{\text{goal}}) > \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ and Γ does not contain a path $\pi^\Gamma(s_{\text{start}}, s_{\text{goal}})$ such that $c^*(\pi^\Gamma(s_{\text{start}}, s_{\text{goal}})) = c^*(s_{\text{start}}, s_{\text{goal}})$. This means that there must exist a pair of states $s_i \in \Gamma$ and $s_{i+1} \in \Gamma$ such that $s_{i+1} \in SUCCS(s_i)$, both $s_i, s_{i+1} \in \pi_{opt}(s_{\text{start}}, s_{\text{goal}})$ and s_i has been expanded but s_{i+1} has not. This is so because at least s_{start} gets expanded whenever $s_{\text{start}} \neq s_{\text{goal}}$.

At the time R* terminates, $k(s_{\text{goal}}) = [1, g(s_{\text{goal}})]$, because otherwise $g(s_{\text{goal}}) \leq \epsilon\, h(s_{\text{start}}, s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ according to lemma 6.

Since the while loop terminated while s_{i+1} was still in $OPEN$, it must have been the case that $k(s_{\text{goal}}) < k(s_{i+1})$. Thus, $k(s_{i+1}) = [1, g(s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}})]$ and consequently,

$$
\begin{aligned}
k(s_{\text{goal}}) &< k(s_{i+1}) \\
g(s_{\text{goal}}) &< g(s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}})
\end{aligned}
$$

According to theorem 1, either $g(s_{i+1}) \leq \epsilon\, h(s_{\text{start}}, s_{i+1})$ or $g(s_{i+1}) = \min_{s' | s_{i+1} \in SUCCS(s')}(v(s') + c_{low}(path_{s', s_{i+1}}))$ (or both). In the former case we get the following contradiction,

$$
\begin{aligned}
g(s_{\text{goal}}) &< \epsilon\, (h(s_{\text{start}}, s_{i+1}) + h(s_{i+1}, s_{\text{goal}})) \\
&\leq \epsilon\, (c^*(s_{\text{start}}, s_{i+1}) + c^*(s_{i+1}, s_{\text{goal}})) \\
&= \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})
\end{aligned}
$$

In the latter case we get a similar contradiction,

$$
\begin{aligned}
g(s_{\text{goal}}) &< \min_{s'|s_{i+1} \in SUCCS(s')}(v(s') + c_{low}(path_{s',s_{i+1}})) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq v(s_i) + c_{low}(path_{s_i,s_{i+1}}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&= g(s_i) + c_{low}(path_{s_i,s_{i+1}}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq g(s_i) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i)) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&= \epsilon\, c^*(s_{\text{start}}, s_i) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&= \epsilon\, c^*(s_{\text{start}}, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq \epsilon\, (c^*(s_{\text{start}}, s_{i+1}) + c^*(s_{i+1}, s_{\text{goal}})) \\
&= \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})
\end{aligned}
$$

■

5 Analysis of the Confidence on ϵ-suboptimality

For the purpose of this analysis, we will assume that all edges in the graph have unit costs and each state s has M states lying at distance Δ edges from it. We will also assume that every time K states are generated by search on line 23, they have not been encountered previously by search. This assumption is correct when the search-space is a tree with a single or multiple goal states. The tree model of a search-space has been commonly used for the statistical analysis of A*-like searches [2, 3, 1]. Our assumption is also approximately correct if K is negligibly small in comparison to the number of states that lie at distance Δ from state s that is being expanded.

Let us introduce the following tree, denoted by Γ^M. The root of the tree is s_{start} and the successors of each node s' in the tree are all M successors lying at distance Δ edges from state s' in the original graph. Note that Γ^M may have two states that are the same state in the original graph but are treated as two different states in Γ^M, because of the way it is constructed. More generally, two states that have different predecessors in Γ^M are treated as different states independently of whether they are really the same state in the original graph.

We define $N_{l,\epsilon}$ to be the number of distinct paths $\pi^{\Gamma^M}(s_{\text{start}}, v)$ in Γ^M such that they satisfy two conditions: (a) a goal state lies within Δ edges from v in the original graph and (b) $c^*(\pi^{\Gamma^M}(s_{\text{start}}, v)) + c^*(v, s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$. Note that any path $\pi^{\Gamma^M}(s_{\text{start}}, v)$ is given as a sequence of states from the tree Γ^M: $\{s_0 = s_{\text{start}}, s_1, \ldots, s_k = v\}$ such that $s_i \in \Gamma^M$ for $0 \leq i \leq k$ and s_i is a predecessor of s_{i+1} in Γ^M for $0 \leq i \leq k-1$. Two paths $\pi_1^{\Gamma^M}(s_{\text{start}}, v)$ and $\pi_2^{\Gamma^M}(s_{\text{start}}, v)$ are considered to be distinct if the sequence of states that appear on these paths differ from each other in any way.

We define a K random walk on any graph G starting with any state s_{start} as a process of iteratively building a tree Γ^K of depth m in the following way: its root is state s_{start}; the successors of any state $s' \in \Gamma^K$ are K randomly selected successors of state s' in G. ith step of a K random walk is defined to be a process of generating all states in Γ^K that will reside at the depth of i from the root of Γ^K. Thus, after the 0th step of the K random walk, Γ^K consists of only s_{start}; after the 1st step of the K random walk, Γ^K consists of s_{start} and K randomly chosen successors of s_{start} from the graph G; after the 2nd step, Γ^K is grown further to contain an additional K^2 states, that are randomly chosen K successors of K states added in the previous step.

The first two theorems are independent of the algorithm proposed in our paper but will be used in the proofs of subsequent theorems.

Theorem 7 *Consider a tree with constant branching factor of M and N_l goal states at depth l distributed uniformly. A K random walk starting at the root s_{start} of this tree generates at least one goal state s_{goal} at lth step with the probability of* $1 - \prod_{i=0}^{N_l-1} \frac{M^l - K^l - i}{M^l - i}$ *if* $N_l \leq M^l - K^l$ *and* 1 *otherwise.*

Proof: Suppose l steps of the K random walk construct a tree Γ^K which is clearly a subset of the original tree. Let t be the number of goal states at level l (leaves) of Γ^K. Then $P(t > 0) = 1 - P(t = 0)$, where $P(t = 0)$ is the probability that Γ^K does not have any goal states as leaves. Γ^K has K^l leaves, whereas the original tree has M^l states at level l and out of them there are N_l goal states. We now will derive a formula for $P(t = 0)$.

Let us use L to denote the set of leaf states of Γ^K and G to denote the set of goal states at level l of the original tree. The event $t = 0$ then corresponds to the event of having $L \cap G = \emptyset$. Suppose first that $N_l > M^l - K^l$. Then the event $L \cap G = \emptyset$ is impossible since N_l is larger than the number of leaves in the original tree, M^l, minus the number of leaves in Γ^K, K^l, meaning that at least one of the goal states will have to be a leaf in Γ^K. Consequently, $P(t > 0) = 1 - P(t = 0) = 1 - 0 = 1$, which is consistent with the theorem.

Now suppose that $N_l \leq M^l - K^l$. Then, $P(t = 0) = \sum_{L,G\, s.t.\, L \cap G = \emptyset} P(L \wedge G) = \sum_{L,G\, s.t.\, L\cap G=\emptyset} P(G|L)P(L)$. Since we assume that goal states at depth l are distributed uniformly, all configurations of N_l goal states at level l are equally probable and therefore $P(G|L) = P(G) = 1/\binom{M^l}{N_l}$. Thus,

$$\begin{aligned}
P(t=0) &= \sum_{L,G\, s.t.\, L\cap G=\emptyset} \frac{1}{\binom{M^l}{N_l}} P(L) &&= \frac{1}{\binom{M^l}{N_l}} \sum_{L,G\, s.t.\, L\cap G=\emptyset} P(L) \\
&= \frac{1}{\binom{M^l}{N_l}} \sum_L \sum_{G\, s.t.\, L\cap G=\emptyset} P(L) &&= \frac{1}{\binom{M^l}{N_l}} \sum_L \binom{M^l-K^l}{N_l} P(L) \\
&= \frac{\binom{M^l-K^l}{N_l}}{\binom{M^l}{N_l}} \sum_L P(L) &&= \frac{\binom{M^l-K^l}{N_l}}{\binom{M^l}{N_l}} \\
&= \prod_{i=0}^{N_l-1} \frac{M^l-K^l-i}{M^l-i}
\end{aligned}$$

(The formula for $P(t = 0)$ above actually corresponds to hypergeometric distribution: the probability that by selecting at random N_l goal states out of M^l states, none of K^l leaves of Γ^K are selected.) From this, the theorem follows by computing $P(t > 0) = 1 - P(t = 0)$. ■

Theorem 8 *Consider a tree with constant branching factor of M and $N_{\leq l}$ goal states distributed uniformly in between levels m and l (including the levels m and l) of the tree. A K random walk starting at the root s_{start} of this tree generates at least one goal state s_{goal} at less than or equal to l steps with the probability of 1 if $K = M$ and $N_{\leq l} > 0$, the probability of at least $1 - e^{-\frac{K^l}{l-m+1}}$ if $K < M$ and $N_{\leq l} > M^l$, and the probability of at least $\min(1 - \prod_{i=0}^{N_{\leq l}-1} \frac{M^l-K^l-i}{M^l-i}, 1 - e^{-\frac{K^{l-1}(M-K)}{l-m}})$ otherwise.*

Proof:

Let $P_{\leq l}(t > 0)$ denote the probability that the K random walk generates at least one goal state at level l or before and $P_{\leq l}(t = 0) = 1 - P_{\leq l}(t > 0)$ denote the probability that none of the goal states are generated by it.

Let us first write out the formula for $P_{\leq l}(t = 0)$. Since goal states are assumed to be distributed uniformly, the probabilities for generating goal states at each level are completely independent. Thus, considering the fact that goal states lie at levels m to l, we get

$$P_{\leq l}(t=0) = P(\text{no goal states generated at level m})*$$
$$P(\text{no goal states generated at level m+1}) * \ldots *$$
$$P(\text{no goal states generated at level } l)$$

We will examine four possibilities: (a) $K = M$ and $N_{\leq l} > 0$, (b) $K = M$ and $N_{\leq l} = 0$, (c) $K < M$ and $N_{\leq l} \leq M^l$ and (d) $K < M$ and $N_{\leq l} > M^l$.

Let us first consider case (a): $K = M$ and $N_{\leq l} > 0$. Because of the latter condition, there exists level j, $m \leq j \leq l$ such that the number of goal states at this level, N_j, is non-zero. Then, $N_j > 0 = M^j - M^j = M^j - K^j$ and therefore, according to theorem 7, P(no goal states generated at level j) $=$ $1 - P$(at least one goal state generated at level j) $= 1 - 1 = 0$. Hence, $P_{\leq l}(t > 0) = 1 - P_{\leq l}(t = 0) = 1 - 0 = 1$, which is consistent with the statement of the theorem.

Let us now consider case (b): $K = M$ and $N_{\leq l} = 0$. Then, clearly the probability of generating one or more states at any level less than or equal to l is 0. That is, $P_{\leq l}(t > 0) = 0$. This is consistent with the theorem, according to which, $P_{\leq l}(t > 0) \geq \min(1-\prod_{i=0}^{N_{\leq l}-1} \frac{M^l - K^l - i}{M^l - i}, 1-e^{-\frac{K^{l-1}(M-K)}{l-m}}) = 1-1 = 0$.

Let us now consider case (c): $K < M$ and $N_{\leq l} \leq M^l$. This falls into the last case of our theorem, and therefore we need to prove that $P_{\leq l}(t > 0) \geq \min(1-\prod_{i=0}^{N_{\leq l}-1} \frac{M^l - K^l - i}{M^l - i}, 1-e^{-\frac{K^{l-1}(M-K)}{l-m}})$, or alternatively, $P_{\leq l}(t = 0) \leq \max(\prod_{i=0}^{N_{\leq l}-1} \frac{M^l - K^l - i}{M^l - i}, e^{-\frac{K^{l-1}(M-K)}{l-m}})$.

We distinguish two scenarios. First, suppose there exists some level j, $m \leq j \leq l$, such that the number of goal states at this level, $N_j > M^j - K^j$. Then, according to theorem 7, P(at least one goal state generated at level j) $= 1$. Therefore, $P_{\leq l}(t = 0) = 0$, which is no more than $\max(\prod_{i=0}^{N_{\leq l}-1} \frac{M^l - K^l - i}{M^l - i}, e^{-\frac{K^{l-1}(M-K)}{l-m}})$ since the second term is positive (in fact, the first term is also guaranteed to be non-negative because the whole product is equal to zero whenever $N_{\leq l} > M^l - K^l$).

Now suppose that at each level j, $m \leq j \leq l$, the number of goal states $N_j \leq M^j - K^j$. Then, according to theorem 7,

$$
\begin{aligned}
P_{\leq l}(t=0) &= P(\text{no goal states generated at level m})* \\
&\quad P(\text{no goal states generated at level m+1}) * \ldots * \\
&\quad P(\text{no goal states generated at level } l) \\
&= \prod_{i=0}^{N_0-1} \frac{M^m-K^m-i}{M^m-i} * \\
&\quad \prod_{i=0}^{N_1-1} \frac{M^{m+1}-K^{m+1}-i}{M^{m+1}-i} * \ldots * \\
&\quad \prod_{i=0}^{N_l-1} \frac{M^l-K^l-i}{M^l-i} \\
&= \prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^j-K^j-i}{M^j-i}
\end{aligned}
$$

Now let us write out $\prod_{i=0}^{N_{\leq l}-1} \frac{M^l-K^l-i}{M^l-i}$ using the fact that $N_{\leq l} = \sum_{j=m}^{l} N_j$:

$$
\begin{aligned}
\prod_{i=0}^{N_{\leq l}-1} \frac{M^l-K^l-i}{M^l-i} &= \prod_{i=0}^{\sum_{j=m}^{l} N_j-1} \frac{M^l-K^l-i}{M^l-i} \\
&= \prod_{i=0}^{N_l-1} \frac{M^l-K^l-i}{M^l-i} * \\
&\quad \prod_{i=N_l}^{N_l+N_{l-1}-1} \frac{M^l-K^l-i}{M^l-i} * \\
&\quad \prod_{i=N_l+N_{l-1}}^{N_l+N_{l-1}+N_{l-2}-1} \frac{M^l-K^l-i}{M^l-i} * \ldots * \\
&\quad \prod_{i=\sum_{t=m+1}^{l} N_t}^{\sum_{t=m}^{l} N_t-1} \frac{M^l-K^l-i}{M^l-i} \\
&= \prod_{j=m}^{l} \prod_{i=\sum_{t=j+1}^{l} N_t}^{\sum_{t=j}^{l} N_t-1} \frac{M^l-K^l-i}{M^l-i} \\
&= \prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^l-K^l-i-\sum_{t=j+1}^{l} N_t}{M^l-i-\sum_{t=j+1}^{l} N_t}
\end{aligned}
$$

We thus need to show that

$$
\prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^j-K^j-i}{M^j-i} \leq \max(\prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^l-K^l-i-\sum_{t=j+1}^{l} N_t}{M^l-i-\sum_{t=j+1}^{l} N_t}, e^{-\frac{K^{l-1}(M-K)}{l-m}}) \tag{2}
$$

We prove by contradiction. Suppose the inequality 2 does not hold. That is,

$$
P_{\leq l}(t=0) = \prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^j-K^j-i}{M^j-i} > \prod_{j=m}^{l} \prod_{i=0}^{N_j-1} \frac{M^l-K^l-i-\sum_{t=j+1}^{l} N_t}{M^l-i-\sum_{t=j+1}^{l} N_t} \tag{3}
$$

and

$$P_{\leq l}(t=0) = \prod_{j=m}^{l} \prod_{i=0}^{N_j - 1} \frac{M^j - K^j - i}{M^j - i} > e^{-\frac{K^{l-1}(M-K)}{l-m}} \tag{4}$$

From inequality 3 follows that there exists some j, i such that $m \leq j \leq l$, $j \leq i \leq N_j - 1$ and

$$\frac{M^j - K^j - i}{M^j - i} > \frac{M^l - K^l - i - \sum_{t=j+1}^{l} N_t}{M^l - i - \sum_{t=j+1}^{l} N_t} \tag{5}$$

Then

$$\begin{array}{lll}
\frac{M^j - K^j - i}{M^j - i} & > & \frac{M^l - K^l - i - \sum_{t=j+1}^{l} N_t}{M^l - i - \sum_{t=j+1}^{l} N_t} \\
1 - \frac{K^j}{M^j - i} & > & 1 - \frac{K^l}{M^l - i - \sum_{t=j+1}^{l} N_t} \\
\frac{K^j}{M^j - i} & < & \frac{K^l}{M^l - i - \sum_{t=j+1}^{l} N_t} \\
K^j & < & \frac{K^l\,(M^j - i)}{M^l - i - \sum_{t=j+1}^{l} N_t} \qquad // M^j - i > 0 \text{ from } i \leq N_j - 1 \leq M^j - K^j - 1 < M^j
\end{array}$$

The term $M^l - i - \sum_{t=j+1}^{l} N_t$ is also strictly positive because

$$\begin{array}{lllll}
M^l - i - \sum_{t=j+1}^{l} N_t & \geq & M^l - N_j + 1 - \sum_{t=j+1}^{l} N_t & = & M^l - \sum_{t=j}^{l} N_t + 1 \\
 & \geq & M^l - \sum_{t=m}^{l} N_t + 1 & = & M^l - N_{\leq l} + 1 \\
 & \geq & M^l - M^l + 1 & > & 0
\end{array}$$

Consequently,

$$\begin{array}{lll}
K^j & < & \frac{K^l\,(M^j - i)}{M^l - i - \sum_{t=j+1}^{l} N_t} \\
K^j\,(M^l - i - \sum_{t=j+1}^{l} N_t) & < & K^l\,(M^j - i) \\
K^j\,M^l - K^j\,i - K^j\,\sum_{t=j+1}^{l} N_t & < & K^l\,M^j - K^l\,i \\
K^j\,M^l - K^j\,i - K^j\,\sum_{t=j+1}^{l} N_t - K^l\,M^j + K^l\,i & < & 0 \\
K^j\,M^l - K^l\,M^j + K^l\,i - K^j\,i - K^j\,\sum_{t=j+1}^{l} N_t & < & 0
\end{array}$$

Since $K^l \geq K^j$ for any j such that $m \leq j \leq l$, we have $K^l\, i - K^j\, i \geq 0$ and therefore,

$$\begin{aligned} K^j\, M^l - K^l\, M^j - K^j \textstyle\sum_{t=j+1}^{l} N_t &< 0 \\ M^l - K^{l-j}\, M^j &< \textstyle\sum_{t=j+1}^{l} N_t \end{aligned} \tag{6}$$

The last line in inequality 6 clearly does not hold for $j = l$. We therefore assume that $m \leq j \leq l-1$. As a result,

$$\begin{aligned} \textstyle\sum_{t=j+1}^{l} N_t &> M^l\,(1 - (\tfrac{K}{M})^{l-j}) \\ &\geq M^l\,(1 - (\tfrac{K}{M})^{l-(l-1)}) \\ &= M^l\,(1 - \tfrac{K}{M}) \end{aligned} \tag{7}$$

This means that there must exist s, $m \leq s \leq l-1$, such that $N_s > \frac{M^l}{l-m}\,(1 - \frac{K}{M})$. Otherwise, $\sum_{t=j+1}^{l} N_t \leq (l-j)\frac{M^l}{l-m}\,(1 - \frac{K}{M}) \leq (l-m)\,\frac{M^l}{l-m}\,(1 - \frac{K}{M}) = M^l\,(1 - \frac{K}{M})$.

Then, according to theorem 7, $P(\text{no goal states generated at level } s)$ is equal to 0 if $N_s > M^s - K^s$ and otherwise:

$$\begin{aligned} P(\text{no goal states generated at level } s) &= \textstyle\prod_{i=0}^{N_s-1} \frac{M^s-K^s-i}{M^s-i} \\ &\leq \textstyle\prod_{i=0}^{\frac{M^l}{l-m}(1-\frac{K}{M})-1} \frac{M^s-K^s-i}{M^s-i} \\ &= \textstyle\prod_{i=0}^{\frac{M^l}{l-m}(1-\frac{K}{M})-1} (1 - \frac{K^s}{M^s-i}) \end{aligned}$$

Since $\frac{M^l}{l-m}\,(1 - \frac{K}{M}) < N_s \leq M^s - K^s$, the variable i can not be larger than or equal to M^s. Therefore,

$$\begin{aligned} 0 < M^s - i &\leq M^s \\ \tfrac{K^s}{M^s-i} &\geq \tfrac{K^s}{M^s} \\ 1 - \tfrac{K^s}{M^s-i} &\leq 1 - \tfrac{K^s}{M^s} \end{aligned}$$

Thus,

$$
\begin{aligned}
P(\text{no goal states generated at level } s) &\leq \prod_{i=0}^{\frac{M^l}{l-m}(1-\frac{K}{M})-1} (1 - \frac{K^s}{M^s - i}) \\
&\leq \prod_{i=0}^{\frac{M^l}{l-m}(1-\frac{K}{M})-1} (1 - (\frac{K}{M})^s) \\
&\leq \prod_{i=0}^{\frac{M^l}{l-m}(1-\frac{K}{M})-1} (1 - (\frac{K}{M})^{l-1}) \\
&= (1 - (\frac{K}{M})^{l-1})^{\frac{M^l}{l-m}(1-\frac{K}{M})} \\
&= e^{\frac{M^l}{l-m}(1-\frac{K}{M})\, ln(1-(\frac{K}{M})^{l-1})}
\end{aligned}
$$

The last line was obtained by taking a logarithm and then exponent of the right hand side. Now let us apply Taylor series expansion to the term $ln(1 - (\frac{K}{M})^{l-1})$. If we let x denote the fraction $-(\frac{K}{M})^{l-1}$, then, as long as $K < M$, $ln(1 + x)$ can be re-written as:

$$
ln(1 + x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots
$$

Moreover, because x is negative, all the terms on the right hand side are negative. Therefore,

$$
\begin{aligned}
ln(1 + x) &\leq x \\
ln(1 - (\frac{K}{M})^{l-1}) &\leq -(\frac{K}{M})^{l-1} \\
e^{\frac{M^l}{l-m}(1-\frac{K}{M})\, ln(1-(\frac{K}{M})^{l-1})} &\leq e^{\frac{M^l}{l-m}(1-\frac{K}{M})(-(\frac{K}{M})^{l-1})} \\
e^{\frac{M^l}{l-m}(1-\frac{K}{M})\, ln(1-(\frac{K}{M})^{l-1})} &\leq e^{-\frac{K^{l-1}(M-K)}{l-m}}
\end{aligned}
$$

Thus, $P(\text{no goal states generated at level } s) \leq e^{-\frac{K^{l-1}(M-K)}{l-m}}$. And since $P_{\leq l}(t = 0) \leq P(\text{no goal states generated at level } s)$ we get $P_{\leq l}(t = 0) \leq e^{-\frac{K^{l-1}(M-K)}{l-m}}$, which contradicts inequality 4.

Finally, let us now consider case (d): $K < M$ and $N_{\leq l} > M^l$. We need to prove that $P_{\leq l}(t > 0) \geq 1 - e^{-\frac{K^l}{l-m+1}}$, or alternatively, $P_{\leq l}(t = 0) \leq e^{-\frac{K^l}{l-m+1}}$.

The fact that $N_{\leq l} = \sum_{t=m}^{l} N_t > M^l$ means that there must exist s, $m \leq s \leq l$, such that $N_s > \frac{M^l}{l-m+1}$.

Then, according to theorem 7, P(no goal states generated at level s) is equal to 0 if $N_s > M^s - K^s$ and otherwise:

$$\begin{aligned} P(\text{no goal states generated at level } s) &= \prod_{i=0}^{N_s-1} \frac{M^s-K^s-i}{M^s-i} \\ &\leq \prod_{i=0}^{\frac{M^l}{l-m+1}-1} \frac{M^s-K^s-i}{M^s-i} \\ &= \prod_{i=0}^{\frac{M^l}{l-m+1}-1} (1-\frac{K^s}{M^s-i}) \end{aligned}$$

Since $\frac{M^l}{l-m+1} < N_s \leq M^s - K^s$, the variable i can not be larger than or equal to M^s. Therefore, as we have shown before $1 - \frac{K^s}{M^s-i} \leq 1 - \frac{K^s}{M^s}$, and thus,

$$\begin{aligned} P(\text{no goal states generated at level } s) &\leq \prod_{i=0}^{\frac{M^l}{l-m+1}-1} (1-\frac{K^s}{M^s-i}) \\ &\leq \prod_{i=0}^{\frac{M^l}{l-m+1}-1} (1-(\frac{K}{M})^s) \\ &\leq \prod_{i=0}^{\frac{M^l}{l-m+1}-1} (1-(\frac{K}{M})^l) \\ &= (1-(\frac{K}{M})^l)^{\frac{M^l}{l-m+1}} \\ &= e^{\frac{M^l}{l-m+1} ln(1-(\frac{K}{M})^l)} \end{aligned}$$

The last line was obtained again by taking a logarithm and then exponent of the right hand side. Once again we apply Taylor series expansion to the term $ln(1+x)$, where x now is $-(\frac{K}{M})^l$ and is less than 1 in magnitude. As before, x is negative and therefore $ln(1+x) \leq x$. As a result,

$$\begin{aligned} ln(1-(\frac{K}{M})^l) &\leq -(\frac{K}{M})^l \\ e^{\frac{M^l}{l-m+1} ln(1-(\frac{K}{M})^l)} &\leq e^{\frac{M^l}{l-m+1} (-(\frac{K}{M})^l)} \\ e^{\frac{M^l}{l-m+1} ln(1-(\frac{K}{M})^l)} &\leq e^{-\frac{K^l}{l-m+1}} \end{aligned}$$

Thus, P(no goal states generated at level s) $\leq e^{-\frac{K^l}{l-m+1}}$. And since $P_{\leq l}(t=0) \leq P$(no goal states generated at level s) we get $P_{\leq l}(t=0) \leq e^{-\frac{K^l}{l-m+1}}$.

■

Theorem 9 *The probability that a particular run of R^* results in a path whose cost is no more than ϵ^2 times the cost l of an optimal path (that is, $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon^2 c^*(s_{\text{start}}, s_{\text{goal}})$) is 1 if $K = M$ or $l \leq \Delta$. Otherwise, it is at least $1 - e^{-\frac{K^H}{H-L+2}}$ if $N_{l,\epsilon} > M^H$ and $\min(1 - \prod_{i=0}^{N_{l,\epsilon}-1} \frac{M^H - K^H - i}{M^H - i}, 1 - e^{-\frac{K^{H-1}(M-K)}{H-L+1}})$ if $N_{l,\epsilon} \leq M^H$, where $L = \lfloor \frac{l}{\Delta} \rfloor$ and $H = \lfloor \frac{\epsilon l}{\Delta} \rfloor$.*

Proof: The case of $K = M$ follows directly from theorem 6.

Now let us consider the case $l \leq \Delta$. First, let us show that s_{start} gets expanded. This is so because during the first while loop test on line 17 s_{start} is the only state in *OPEN* and therefore the only reason for it not to have been expanded is if the test failed. This is impossible, however, since during the test $\min_{s' \in OPEN} k(s') = k(s_{\text{start}}) = [0; \epsilon\, h(s_{\text{start}}, s_{\text{goal}})]$ and $k(s_{\text{goal}}) = [1; \infty]$ if $s_{\text{start}} \neq s_{\text{goal}}$ and $k(s_{\text{goal}}) = k(s_{\text{start}})$ otherwise. Thus, it must be the case that $k(s_{\text{goal}}) \geq \min_{s' \in OPEN} k(s')$.

Now, because s_{start} is guaranteed to be expanded, s_{goal} is generated on line 25 during the first expansion, namely the expansion of s_{start}. R* terminates when either $k(s_{\text{goal}}) < \min_{s' \in OPEN} k(s')$ or $OPEN = \emptyset$. Therefore, s_{goal} must be expanded at the time R* terminates. Consequently, from theorem 5, it follows that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq g(s_{\text{goal}}) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{\text{goal}})) = \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}}) \leq \epsilon^2 c^*(s_{\text{start}}, s_{\text{goal}})$.

To prove the other cases of the theorem we need to derive a lower bound on the probability $P(c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon^2 c^*(s_{\text{start}}, s_{\text{goal}}))$. To derive this bound, we actually compute a lower bound on the probability that the graph Γ contains at least one path $\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})$ such that $c^*(\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$. This bound will result in the required bound because according to theorem 5 $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_{\text{goal}}))$ and therefore $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon\, c^*(\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon^2 c^*(s_{\text{start}}, s_{\text{goal}})$.

The derivation of the lower bound on the probability that the graph Γ contains at least one path $\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})$ such that $c^*(\pi^{\Gamma}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$, consists of several steps.

First, consider a path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ in Γ^M tree from state s_{start} to state s such that it satisfies two conditions: (a) a goal state lies within Δ edges from s in the original graph; (b) $c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + c^*(s, s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$. The latter condition is equivalent to saying that the cost of the path from

s_{start} to s_{goal} that passes through the states in Γ^M in the same order they appear in the path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ and follows an optimal path in between any two consecutive states in Γ^M and from s to s_{goal} is no more than ϵ times the cost of an optimal path from s_{start} to s_{goal} in the original graph. The number of such paths is $N_{l,\epsilon}$, according to the definition of $N_{l,\epsilon}$.

Now let us analyze the following tree Γ' of depth : the root of the tree is s_{start}; for each non-leaf state s in Γ', its successors are the same as in Γ, generated at random on line 23, if s was expanded by R*, and its successors are a new set of successors generated according to line 23, if s was not expanded by R*. The process of generating Γ' is a K random walk on the tree Γ^M. We will first compute a lower bound on the probability that Γ' contains at least one of the paths $\pi^{\Gamma^M}(s_{\text{start}}, s)$ satisfying conditions (a) and (b). Because of the condition (b), any such path terminates at level that is somewhere in between $L-1$ and H. The path must terminate at the level higher than or equal to $L-1$ because any path that terminates at the level lower than $L-1$ will have $c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) < \Delta\,(\lfloor \frac{l}{\Delta} \rfloor - 1) \leq l - \Delta$. Therefore, s_{goal} can not lie within Δ edges from s for otherwise $c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + c^*(s, s_{\text{goal}})$ would have been less than l which is already the cost of an optimal path from s_{start} to s_{goal}. The path must also terminate at the level that is smaller than or equal to H because any path that terminates at the level higher than H will have $c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) \geq \Delta\,(\lfloor \frac{\epsilon l}{\Delta} \rfloor + 1) > \epsilon\, l - \Delta + \Delta = \epsilon\, l$. This contradicts the condition (b).

Thus, any path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ satisfying conditions (a) and (b) terminates at level that is somewhere in between $L-1$ and H. Therefore, given our assumption, that during the search we do not encounter previously generated samples, according to theorem 8, a lower bound on the probability that Γ' contains one of such paths can be given as $1 - e^{-\frac{K^H}{H-L+2}}$ if $N_{l,\epsilon} > M^H$ and $\min(1 - \prod_{i=0}^{N_{l,\epsilon}-1} \frac{M^H - K^H - i}{M^H - i}, 1 - e^{-\frac{K^{H-1}(M-K)}{H-L+1}})$ if $N_{l,\epsilon} \leq M^H$.

We now show that that if the tree Γ' contains one or more of paths $\pi^{\Gamma^M}(s_{\text{start}}, s)$ satisfying conditions (a) and (b), then the path returned by R* is ϵ^2 suboptimal. Consequently, the lower bound for the tree Γ' is also a lower bound for R* to find an ϵ^2 suboptimal solution.

We prove by contradiction. Suppose Γ' contains one or more of paths $\pi^{\Gamma^M}(s_{\text{start}}, s) = \{s_0 = s_{\text{start}}, s_1, \ldots s_i, \ldots, s_k = s\}$ satisfying conditions (a) and (b), but the path returned by R* is not ϵ^2 suboptimal. That is, at the time R* terminates $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) > \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}})$.

Let us now consider a pair s_i, s_{i+1} from the path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ such that s_i has been expanded but s_{i+1} has not been expanded. First, let us show that such pair must exist. That is, it is impossible for s_{start} not to have been expanded and it is also impossible for all the states on the path to have been expanded. The former scenario is impossible because during the first while loop test on line 17 s_{start} is the only state in *OPEN* and therefore the only reason for it not to have been expanded is if the test failed. This is impossible, however, since during the test $\min_{s' \in OPEN} k(s') = k(s_{\text{start}}) = [0; \epsilon\, h(s_{\text{start}}, s_{\text{goal}})]$ and $k(s_{\text{goal}}) = [1; \infty]$ if $s_{\text{start}} \neq s_{\text{goal}}$ and $k(s_{\text{goal}}) = k(s_{\text{start}})$ otherwise. Thus, it must be the case that $k(s_{\text{goal}}) \geq \min_{s' \in OPEN} k(s')$.

It is also impossible for all the states on the path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ to have been expanded for the following reasons. If $s_{\text{goal}} = s_{\text{start}}$ then $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) = 0 = \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}})$ which contradicts our initial assumption. If $s_{\text{goal}} \neq s_{\text{start}}$, then R* would have generated s_{goal} as a successor of s and consequently,

$$
\begin{aligned}
g(s_{\text{goal}}) &= v(s) + c_{low}(path_{s,s_{\text{goal}}}) && //theorem\ 1 \\
&= g(s) + c_{low}(path_{s,s_{\text{goal}}}) && //lemma\ 3 \\
&\leq g(s) + \epsilon\, c^*(s, s_{\text{goal}}) && //lemma\ 1 \\
&\leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s)) + \epsilon\, c^*(s, s_{\text{goal}}) && //theorem\ 5 \\
&\leq \epsilon\, c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + \epsilon\, c^*(s, s_{\text{goal}}) && \\
&= \epsilon\, (c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + c^*(s, s_{\text{goal}})) && \\
&\leq \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}}) && //condition(b)
\end{aligned}
$$

From theorem 2 it then follows that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}})$ which contradicts our initial assumption. Thus, it must be the case that there exists a pair s_i, s_{i+1} from the path $\pi^{\Gamma^M}(s_{\text{start}}, s)$ such that s_i has been expanded but s_{i+1} has not been expanded.

We now consider two cases. First, suppose at the time R* terminates, $k(s_{\text{goal}}) = [0; g(s_{\text{goal}})]$. Then, $s_{\text{goal}} \in$ *CLOSED* since initially $k(s_{\text{goal}}) = [1; g(s_{\text{goal}})]$; whenever a key of a state is modified, the state is inserted into *OPEN*; and $s_{\text{goal}} \notin$ *OPEN* due to the termination condition of the while loop. Thus, at the time s_{goal} was selected for expansion, according to lemma 6, $g(s_{\text{goal}}) \leq \epsilon\, h(s_{\text{start}}, s_{\text{goal}}) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}})$ and from theorem 2 it follows that $c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) \leq \epsilon\, c^*(s_{\text{start}}, s_{\text{goal}}) \leq \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}})$. However, this contradicts our initial assumption since a g-value of a state that was expanded remains constant according to lemma 3.

Now suppose at the time R* terminates, $k(s_{\text{goal}}) = [1; g(s_{\text{goal}})]$. Since the while loop terminated while s_{i+1} was still in $OPEN$, it must have been the case that $k(s_{\text{goal}}) < k(s_{i+1})$. Thus, $k(s_{i+1}) = [1, g(s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}})]$ and consequently,

$$\begin{array}{lcl} k(s_{\text{goal}}) & < & k(s_{i+1}) \\ g(s_{\text{goal}}) & < & g(s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\ c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) & < & g(s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \quad //\textit{theorem } 2 \end{array}$$

According to theorem 1, either $g(s_{i+1}) \leq \epsilon\, h(s_{\text{start}}, s_{i+1})$ or $g(s_{i+1}) = \min_{s'|s_{i+1} \in SUCCS(s')}(v(s') + c_{low}(path_{s',s_{i+1}})) \leq v(s_i) + c_{low}(path_{s_i,s_{i+1}})$ (or both). In the former case we get the following contradiction,

$$\begin{array}{lcl} c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) & < & \epsilon\,(h(s_{\text{start}}, s_{i+1}) + h(s_{i+1}, s_{\text{goal}})) \\ & \leq & \epsilon\,(c^*(\pi^{\Gamma^M}(s_{\text{start}}, s_{i+1})) + c^*(\pi^{\Gamma^M}(s_{i+1}, s)) + c^*(s, s_{\text{goal}})) \\ & = & \epsilon\,(c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + c^*(s, s_{\text{goal}})) \\ & \leq & \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}}) \end{array}$$

In the latter case we get a similar contradiction using theorem 5 and condition (b),

$$
\begin{aligned}
c(\pi^{\Gamma}_{bp}(s_{\text{start}}, s_{\text{goal}})) &< v(s_i) + c_{low}(path_{s_i,s_{i+1}}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&= g(s_i) + c_{low}(path_{s_i,s_{i+1}}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq g(s_i) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq \epsilon\, c^*(\pi^{\Gamma}_{opt}(s_{\text{start}}, s_i)) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq \epsilon\, c^*(\pi^{\Gamma^M}(s_{\text{start}}, s_i)) + \epsilon\, c^*(s_i, s_{i+1}) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&= \epsilon\, c^*(\pi^{\Gamma^M}(s_{\text{start}}, s_{i+1})) + \epsilon\, h(s_{i+1}, s_{\text{goal}}) \\
&\leq \epsilon\, c^*(\pi^{\Gamma^M}(s_{\text{start}}, s_{i+1})) + \epsilon\,(c^*(\pi^{\Gamma^M}(s_{i+1}, s)) + h(s, s_{\text{goal}})) \\
&= \epsilon\,(c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + h(s, s_{\text{goal}})) \\
&\leq \epsilon\,(c^*(\pi^{\Gamma^M}(s_{\text{start}}, s)) + c^*(s, s_{\text{goal}})) \\
&\leq \epsilon^2\, c^*(s_{\text{start}}, s_{\text{goal}})
\end{aligned}
$$

■

References

[1] J. Gaschnig. Performance measurement and analysis of certain search algorithms. Tech. Rep. CMU-CS-79-124, Carnegie Mellon University, 1979.

[2] J. Pearl. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[3] I. Pohl. Practical and theoretical considerations in heuristic search algorithms. In E. W. Elcock and D. Michie, editors, *Machine Intelligence 8*, pages 55–72. New York: Wiley, 1977.