

Chapter S:VI

VI. Relaxed Models

- ❑ Motivation
- ❑ ϵ -Admissible Speedup Versions of A*
- ❑ Using Information about Uncertainty of h
- ❑ Risk Measures

- ❑ Nonadditive Evaluation Functions

- ❑ Heuristics Provided by Simplified Models
- ❑ Mechanical Generation of Admissible Heuristics
- ❑ Probability-Based Heuristics

Motivation

- Optimization problems.

If the available heuristic is an optimistic estimate of h^* , then A^* is guaranteed to find an optimal solution path if one exists.

→ The solution path found by A^* is optimal.

- Constraint satisfaction problems.

If several near-optimum solutions exist, then A^* uniformly follows the different paths, spending a lot of time.

→ The admissibility property becomes a curse rather than a virtue.

Motivation

Basic Questions from Search Theory [Barr/Feigenbaum 1981]

1. Let minimizing effort be more important than minimizing solution cost.

Is $f = g + h$ an appropriate evaluation function in this case?

2. Even if solution cost is important, an admissible search might take too long.

Can speed be gained at the cost of a *bounded* decrease in solution quality?

3. For some problems, all good heuristics ($h \approx h^*$) are not optimistic.

How is the search affected by an inadmissible heuristic function?

Remarks:

- ❑ Up to now, we used the paradigm “small-is-quick”: Focusing the search effort toward finding a smallest solution (e.g., shortest solution path) leads to a smaller search effort in finding a solution.
- ❑ The above observations cast doubt on the appropriateness of the small-is-quick paradigm in satisficing problems. Would it not be better to focus more on nodes which are assumed close to *some* solution?

Motivation

Examination of g and h

Recall that A* orders nodes on OPEN by $f = g + h$.

- g represents the breadth-first component of A* search.
Nodes closer to the start s are preferred.
 - h represents the depth-first component of A* search.
Nodes *estimated to be* closer to a goal γ are preferred.
- We can adjust the balance of the breadth-first and depth-first components for satisficing or semi-optimization problems.

Motivation

Examination of g and h

Recall that A* orders nodes on OPEN by $f = g + h$.

- g represents the breadth-first component of A* search.
Nodes closer to the start s are preferred.
 - h represents the depth-first component of A* search.
Nodes *estimated to be* closer to a goal γ are preferred.
- We can adjust the balance of the breadth-first and depth-first components for satisficing or semi-optimization problems.

Adding weights to the components of f [Pohl 1970]:

$$f_w(n) = (1 - w) \cdot g(n) + w \cdot h(n) \quad \text{with } w \in [0; 1]$$

- $w = 0 \rightsquigarrow$ Uniform-cost search
- $w = \frac{1}{2} \rightsquigarrow$ A*
- $w = 1 \rightsquigarrow$ BF* with $f = h$.

Remarks:

- ❑ 1. For $w \approx 0$, the estimate of the remaining cost is (nearly) ignored.
- ❑ 2. For $w \approx 1$, the current path cost is (nearly) ignored.

In which cases should the first option be preferred, in which cases the second option?

- ❑ For $w \in [0; \frac{1}{2}]$, if h is admissible, then best-first search with f_w is admissible.

But it can be shown that a weighted best-first search with $w \in [0; \frac{1}{2}]$ will expand all nodes n with $h(n) > 0$ that are expanded by A*. Thus it is disadvantageous to use $w < \frac{1}{2}$.

- ❑ For $w \in (\frac{1}{2}; 1]$, even if h is admissible, best-first search with f_w is not admissible in the general case.
- ❑ Usually, the choice $w = 1$ is not adequate. Why?

ϵ -Admissible Speedup Versions of A*

Bounded Decrease in Solution Quality

General Idea

- ❑ Strengthening the depth-first component to find *some* solution faster.
- ❑ Guaranteeing that the cost of the found solution will be *near* the optimal cost.

ε -Admissible Speedup Versions of A*

Bounded Decrease in Solution Quality

General Idea

- Strengthening the depth-first component to find *some* solution faster.
- Guaranteeing that the cost of the found solution will be *near* the optimal cost.

Definition 67 (ε -Admissibility)

An algorithm is called ε -admissible for some $\varepsilon \geq 0$, if – in case solutions exist – it terminates with solution cost C such that

$$C \leq (1 + \varepsilon) \cdot C^*$$

Two approaches:

1. Adjusting the evaluation function in A*: WA*, DWA*.
2. Adjusting the node selection of A* from OPEN: A* _{ε} .

ε -Admissible Speedup Versions of A*

Static Weighting A* Search: WA* [Pohl 1970]

We use the weighting function discussed previously:

$$f_w(n) = (1 - w) \cdot g(n) + w \cdot h(n) \quad \text{with } w \in [0.5; 1]$$

Equivalent formulation (scaling f_w by $\frac{1}{1-w}$):

$$f_\varepsilon(n) = g(n) + (1 + \varepsilon) \cdot h(n) \quad \text{with } \varepsilon > 0$$

BF* using f_ε with $\varepsilon > 0$ is called (static) weighting A* or WA*.

ε -Admissible Speedup Versions of A*

Static Weighting A* Search: WA* [Pohl 1970]

We use the weighting function discussed previously:

$$f_w(n) = (1 - w) \cdot g(n) + w \cdot h(n) \quad \text{with } w \in [0.5; 1]$$

Equivalent formulation (scaling f_w by $\frac{1}{1-w}$):

$$f_\varepsilon(n) = g(n) + (1 + \varepsilon) \cdot h(n) \quad \text{with } \varepsilon > 0$$

BF* using f_ε with $\varepsilon > 0$ is called (static) weighting A* or WA*.

- Using evaluation functions f_ε with $\varepsilon > 0$ in A* does not change path cost calculations (g -part).
- All results for A* that don't require admissible heuristic functions h as a precondition are also valid for WA*.

ε -Admissible Speedup Versions of A*

Static Weighting A* Search: WA* [Pohl 1970]

We use the weighting function discussed previously:

$$f_w(n) = (1 - w) \cdot g(n) + w \cdot h(n) \quad \text{with } w \in [0.5; 1]$$

Equivalent formulation (scaling f_w by $\frac{1}{1-w}$):

$$f_\varepsilon(n) = g(n) + (1 + \varepsilon) \cdot h(n) \quad \text{with } \varepsilon > 0$$

BF* using f_ε with $\varepsilon > 0$ is called (static) weighting A* or WA*.

- Using evaluation functions f_ε with $\varepsilon > 0$ in A* does not change path cost calculations (g -part).
- All results for A* that don't require admissible heuristic functions h as a precondition are also valid for WA*.

ε should be chosen in such a way that $(1 + \varepsilon) \cdot h$ is **not admissible**. Why?

ϵ -Admissible Speedup Versions of A*

Static Weighting A* Search: WA* [Pohl 1970]

Theorem 68 (ϵ -Admissibility of WA*)

Let G be a search space graph with $Prop(G)$ and $\epsilon > 0$. Then WA* with selection function f_ϵ and an admissible heuristic function h is ϵ -admissible.

WA* terminates with solution cost C with $C \leq (1 + \epsilon) \cdot C^*$ if solutions exist.

ϵ -Admissible Speedup Versions of A*

Static Weighting A* Search: WA* [Pohl 1970]

Theorem 68 (ϵ -Admissibility of WA*)

Let G be a search space graph with $Prop(G)$ and $\epsilon > 0$. Then WA* with selection function f_ϵ and an admissible heuristic function h is ϵ -admissible.

WA* terminates with solution cost C with $C \leq (1 + \epsilon) \cdot C^*$ if solutions exist.

Proof (sketch)

1. [Theorem “Completeness”] implies completeness of WA*, since WA* differs from A* only in the evaluation function used and since all restrictions for h are also met by $(1 + \epsilon) \cdot h$.
2. Let WA* terminate with goal node γ and solution cost $C = f_\epsilon(\gamma)$.
3. Let n' be the shallowest OPEN node on some optimal path at termination. Then we have $f_\epsilon(n') = g^*(n') + (1 + \epsilon) \cdot h(n') \leq (1 + \epsilon) \cdot (g^*(n') + h(n'))$.
[Corollary “Shallowest OPEN Node on Optimum Path” also holds for WA*]
4. Since h is admissible, we have $f_\epsilon(n') \leq (1 + \epsilon) \cdot (g^*(n') + h^*(n'))$
5. From $g^*(n') + h^*(n') = C^*$ (node on optimum path) follows that $f_\epsilon(n') \leq (1 + \epsilon) \cdot C^*$.
6. Since WA* selects nodes with smallest f_ϵ -values, we have $C \leq f_\epsilon(n') \leq (1 + \epsilon) \cdot C^*$.

ε -Admissible Speedup Versions of A*

Dynamic Weighting A* Search: DWA* [Pohl 1973]

Idea: Emphasize the depth-first component at the start, but use a balanced weighting near the end to find solutions closer to the optimum:

$$f_{d\varepsilon}(n) = g(n) + \left(1 + \left(1 - \frac{\min(\text{depth}(n), N)}{N} \right) \cdot \varepsilon \right) \cdot h(n)$$

$\text{depth}(n)$: depth of node n (length of backpointer path to n)

N : (anticipated) depth of a desired goal node.

- $\text{depth}(n) \ll N$: h is given a supportive weight equal to $(1 + \varepsilon)$.
 - Depth-first excursions are encouraged.
- $\text{depth}(n)$ near N : Termination is likely to occur.
 - More emphasis on (near) optimality.

BF* using $f_{d\varepsilon}$ with $\varepsilon > 0$ is called dynamic weighting A* or DWA*.

Remarks:

- ❑ For $\varepsilon \rightarrow 0$ we have $f_{(d)\varepsilon}(n) \rightarrow g(n) + h(n)$.
- ❑ Like for WA^* , Corollary “Shallowest OPEN Node on Optimum Path” can be proven analogously for DWA^* .
- ❑ Note that, even if h is monotone, the $f_{d\varepsilon}$ -values can decrease even along an optimum path.
- ❑ Moreover, monotonicity does not longer imply that no nodes are reopened.

ε -Admissible Speedup Versions of A*

Dynamic Weighting A* Search: DWA* [Pohl 1973]

Theorem 69 (ε -Admissibility of DWA*)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Then DWA* with selection function $f_{d\varepsilon}$ and admissible heuristic function h is ε -admissible.

ε -Admissible Speedup Versions of A^*

Dynamic Weighting A^* Search: DWA^* [Pohl 1973]

Theorem 69 (ε -Admissibility of DWA^*)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Then DWA^* with selection function $f_{d\varepsilon}$ and admissible heuristic function h is ε -admissible.

Proof (sketch)

1. Using the same argumentation as for WA^* , we arrive at

$$f_{d\varepsilon}(n') \leq \underbrace{\left(1 + \left(1 - \frac{\min(\text{depth}(n'), N)}{N}\right) \cdot \varepsilon\right)}_{\in [0;1]} \cdot \underbrace{(g^*(n') + h^*(n'))}_{C^*}$$

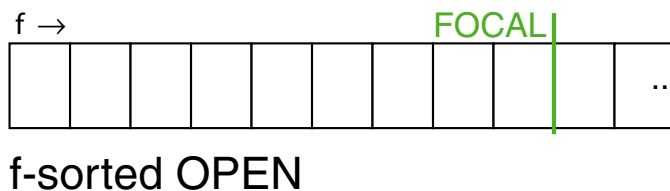
2. Therefore we have $C \leq f_{d\varepsilon}(n') \leq (1 + \varepsilon) \cdot C^*$.

ϵ -Admissible Speedup Versions of A^*

Node Selection by $h_F(n)$: A^*_ϵ [Pearl/Kim 1982]

Idea: Selecting nodes depth-first-like from the cheapest OPEN nodes:

$$\text{FOCAL} = \{n \in \text{OPEN} \mid f(n) \leq (1 + \epsilon) \cdot \min_{n' \in \text{OPEN}} f(n')\}$$



→ Nodes on FOCAL promise (roughly) equal quality solution paths.

- ❑ Instead of selecting the node n on OPEN with smallest $f(n)$ for expansion, we choose the node n' on FOCAL with smallest $h_F(n')$.
- ❑ The function $h_F(n)$ estimates the **computational effort** for completing the search from n .

BF* using $h_F(n)$ on FOCAL for node selection and $\epsilon > 0$ is called A^*_ϵ .

Remarks:

- ❑ Clearly, for $\varepsilon = 0$, A^*_ε reduces to A^* with h_F as a tie-breaker.
- ❑ $h_F(n)$ utilizes knowledge about the problem domain or about the structure of the search space graph (like h).
- ❑ Q. How can the depth-first component of A^* be emphasized using FOCAL and h_F ?
- ❑ A^*_ε uses two heuristic functions: h and h_F .
 - h is used in forming FOCAL. It estimates the best-case reduction in solution quality for the remaining path.
 - h_F is used for selecting nodes from within FOCAL. It estimates the computational effort for the remaining path.
- ❑ The paradigm “small-is-quick” is implemented by $h_F = f = g + h$.

ε -Admissible Speedup Versions of A*

Node Selection by $h_F(n)$: A^*_ε [Pearl/Kim 1982]

Theorem 70 (ε -Admissibility of A^*_ε)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Then A^*_ε with selection from FOCAL and admissible heuristic function h is ε -admissible.

ε -Admissible Speedup Versions of A^*

Node Selection by $h_F(n)$: A^*_ε [Pearl/Kim 1982]

Theorem 70 (ε -Admissibility of A^*_ε)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Then A^*_ε with selection from FOCAL and admissible heuristic function h is ε -admissible.

Proof (sketch)

1. Completeness of A^*_ε can be proven analogously to the proof of completeness of A^* [Theorem “Completeness”] using $(1 + \varepsilon) \cdot M$ as cost bound for paths.
2. Let A^*_ε terminate with goal node γ and solution cost $C = f(\gamma)$.
3. Let n' be the shallowest OPEN node on some optimal path at termination. Then we have $f(n') = g^*(n') + h(n')$. [Corollary “Shallowest OPEN Node on Optimum Path”]
4. Since h is admissible, we have $f(n') \leq g^*(n') + h^*(n')$
5. From $g^*(n') + h^*(n') = C^*$ (node on optimum path) follows that $f(n') \leq C^*$.
6. Let n be the OPEN node with smallest $f(n)$. By definition we have $f(n) \leq f(n')$.
7. Since γ was selected from FOCAL, we have $C \leq f(n) \cdot (1 + \varepsilon)$.
8. Therefore $C \leq f(n') \cdot (1 + \varepsilon)$.
9. Hence $C \leq C^* \cdot (1 + \varepsilon)$.

Remarks:

- A^* and A^*_ε use the same evaluation function $f = g + h$, only the selection rules based on f differ. Hence, all results for A^* that do not rely on the selection rule, e.g. termination on finite graphs, completeness for finite graphs, [Lemma “Shallowest OPEN Node on Path”](#), [Corollary “Shallowest OPEN Node on Optimum Path”](#), and [Lemma “ \$C^*\$ -bounded OPEN Node”](#), can be proven in the same way for A^*_ε .

Completeness for infinite graphs can be proven analogously to the proof for A^* ([Theorem “Completeness”](#)) using bound $(1 + \varepsilon) \cdot M$ instead of M in step 5.

- h_F is allowed to be non-admissible. This does not affect ε -admissibility of A^*_ε .

ϵ -Admissible Speedup Versions of A^*

Comparison of DWA^* and A^*_ϵ

- Advantage of DWA^* :
Easy to implement on basis of A^* .

- Disadvantage of DWA^* :
Depth N of optimal/good solutions has to be estimated a priori.

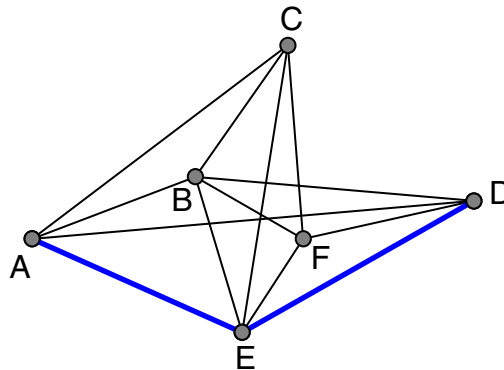
- Advantage of A^*_ϵ :
The separation of the two heuristics h and h_F enables the use of sophisticated estimations of the computational cost, like
 - *global* analysis of the backpointer path from s to n , or
 - utilization of non-additive or non-recursive functions.

ε -Admissible Speedup Versions of A^*

Comparison of DWA^* and A^*_ε

Application of A^* , DWA^* and A^*_ε to Traveling Salesman problems. [Pearl/Kim 1982]

- ❑ 9 cities. Simple TSPs: cities distributed independently and uniformly in the unit square, i.e. distances in $(0; 1.414)$.
“Hard” TSPs: distances independently chosen from a uniform distribution over $(0.75; 1.25)$.
- ❑ A^* , DWA^* and A^*_ε use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j , while i and j range over the *unvisited* cities.
- ❑ DWA^* uses $N = 9$ (search depth is 9), DWA^* and A^*_ε use $\varepsilon \in (0; 0.2]$.
- ❑ The focal-heuristic h_F of A^*_ε is the number of unvisited cities.

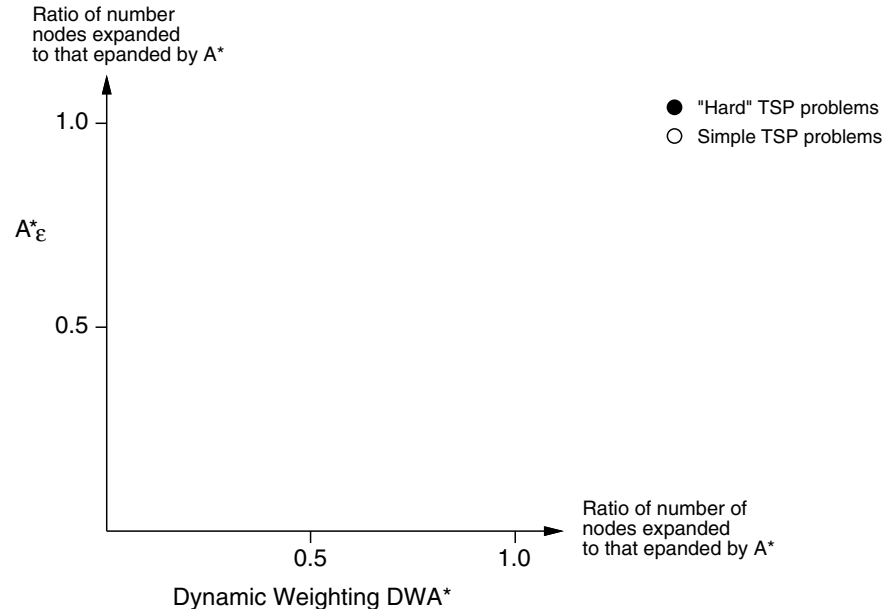


ε -Admissible Speedup Versions of A^*

Comparison of DWA^* and A^*_ε

Application of A^* , DWA^* and A^*_ε to Traveling Salesman problems. [Pearl/Kim 1982]

- ❑ 9 cities. Simple TSPs: cities distributed independently and uniformly in the unit square, i.e. distances in $(0; 1.414)$.
“Hard” TSPs: distances independently chosen from a uniform distribution over $(0.75; 1.25)$.
- ❑ A^* , DWA^* and A^*_ε use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j , while i and j range over the *unvisited* cities.
- ❑ DWA^* uses $N = 9$ (search depth is 9), DWA^* and A^*_ε use $\varepsilon \in (0; 0.2]$.
- ❑ The focal-heuristic h_F of A^*_ε is the number of unvisited cities.

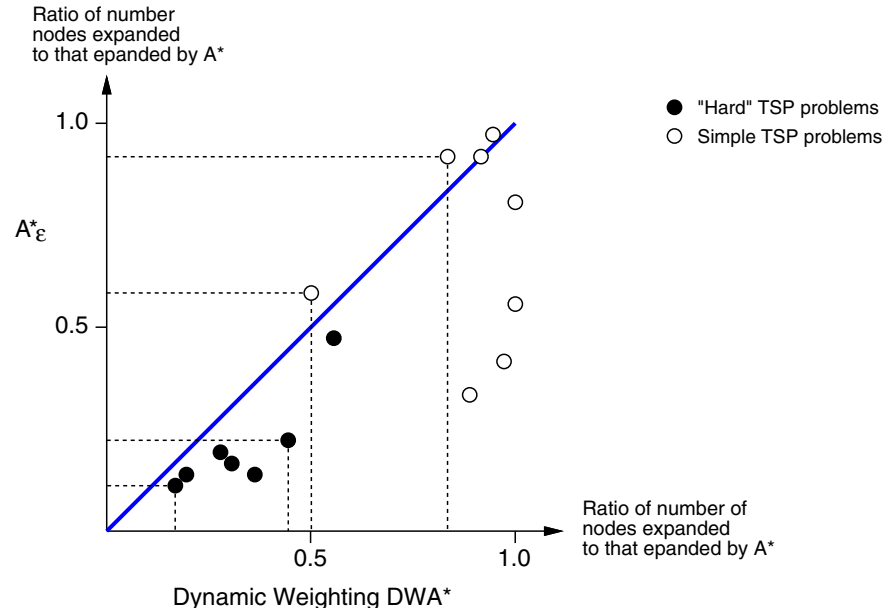


ε -Admissible Speedup Versions of A^*

Comparison of DWA^* and A^*_ε

Application of A^* , DWA^* and A^*_ε to Traveling Salesman problems. [Pearl/Kim 1982]

- ❑ 9 cities. Simple TSPs: cities distributed independently and uniformly in the unit square, i.e. distances in $(0; 1.414)$.
“Hard” TSPs: distances independently chosen from a uniform distribution over $(0.75; 1.25)$.
- ❑ A^* , DWA^* and A^*_ε use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j , while i and j range over the *unvisited* cities.
- ❑ DWA^* uses $N = 9$ (search depth is 9), DWA^* and A^*_ε use $\varepsilon \in (0; 0.2]$.
- ❑ The focal-heuristic h_F of A^*_ε is the number of unvisited cities.



Remarks:

- ❑ Each coordinate represents the ratio of the number of nodes expanded by the corresponding algorithm to that expanded by A^* (with the same heuristic h).
- ❑ The ε -admissible algorithms save computational effort (number of nodes expanded) ranging between 60% and 90% for “hard” TSPs in comparison to A^* .
- ❑ The chart indicates comparable performances for the two algorithms with an advantage for A^*_ε for this (simple) experiment.
- ❑ If the Traveling Salesman problem is applied to a sparsely connected road map, the number of edges in the unexplored portion of the graph would usually constitute a more valid estimation of the remaining computational effort than the proportion of unexplored cities $\left(1 - \frac{\text{depth}(n)}{N}\right)$, which guides the dynamic weighting algorithm.

ϵ -Admissible Speedup Versions of A^*

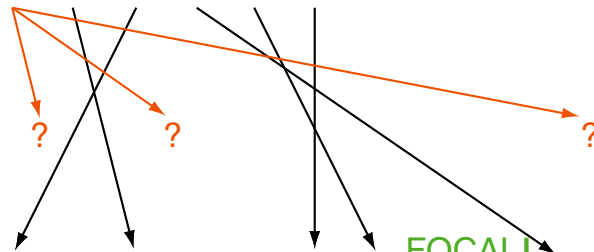
Unifying View: WA^* and DWA^* as variants of A^*_ϵ

Approach: Use $h_F = f_\epsilon$ resp. $h_F = f_{d\epsilon}$ in A^*_ϵ .

$f_{(d)\epsilon}$ -sorted OPEN



(D)WA*



A^*_ϵ

f -sorted OPEN

Problem: Is it guaranteed that $(\operatorname{argmin}_{n \in \text{OPEN}} f_{(d)\epsilon}(n)) \in \text{FOCAL}$ holds?

Remarks:

- When implementing WA^* and DWA^* as variants of A^*_ε , we have to use the same tie breaking strategy for h_F in A^*_ε as was used in $(D)WA^*$ for $f_{(d)\varepsilon}$.

ε -Admissible Speedup Versions of A^*

Lemma 71 (WA^* and DWA^* are variants of A^*_ε)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Further let $f = g + h$ be the usual evaluation function and f' a second evaluation function with

$$f(n) \leq f'(n) \leq (1 + \varepsilon) \cdot f(n) \quad \text{for any } n \in G.$$

Then, for any subset OPEN of nodes in G with $n'_0 := \operatorname{argmin}_{n \in \text{OPEN}} f'(n)$ we have

$$f(n'_0) \leq (1 + \varepsilon) \min_{n \in \text{OPEN}} f(n)$$

ε -Admissible Speedup Versions of A^*

Lemma 71 (WA^* and DWA^* are variants of A^*_ε)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. Further let $f = g + h$ be the usual evaluation function and f' a second evaluation function with

$$f(n) \leq f'(n) \leq (1 + \varepsilon) \cdot f(n) \quad \text{for any } n \in G.$$

Then, for any subset OPEN of nodes in G with $n'_0 := \operatorname{argmin}_{n \in \text{OPEN}} f'(n)$ we have

$$f(n'_0) \leq (1 + \varepsilon) \min_{n \in \text{OPEN}} f(n)$$

Proof (sketch)

Let $n_0 := \operatorname{argmin}_{n \in \text{OPEN}} f(n)$. Then we have

$$\begin{aligned} f(n'_0) &\leq f'(n'_0) \\ &\leq f'(n_0) \\ &\leq (1 + \varepsilon) \cdot f(n_0) \\ &= (1 + \varepsilon) \cdot \min_{n \in \text{OPEN}} f(n) \end{aligned}$$

(Distinguish n_0 and n'_0 resp. f and f' and the chain of inequalities above.)

ε -Admissible Speedup Versions of A*

Pruning Power of h for A^*_ε [A* Condition II]

Corollary 72 (Necessary Condition for Node Expansion II for A^*_ε)

Let G be a search space graph with $Prop(G)$, an admissible heuristic function h , and $\varepsilon > 0$. For any node n expanded by A^*_ε we have a $(1 + \varepsilon) \cdot C^*$ -bounded path from s to n in G .

At time of expansion of a node n we have $f(n) \leq (1 + \varepsilon) \cdot C^*$.

Q. Is there a corresponding sufficient condition for node expansion?

Remarks:

- ❑ This corollary holds also for WA^* and DWA^* (as special cases of A^*_ε).
- ❑ A proof can be given analogously to the proof of Theorem “Necessary Condition for Node Expansion II”.
- ❑ Analogously to Lemma “ C^* -bounded OPEN Node”, it can be proven that at any time before termination there is a node n' on OPEN with $f(n') \leq C^*$.

Therefore, no node n with $f(n) > (1 + \varepsilon) \cdot C^*$ is contained in FOCAL. Hence, such a node n cannot be selected for expansion.

ϵ -Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in A^*_ϵ

When using a monotone heuristic function in A^* ,

- at time of expansion of a node n an optimal path from s to n (the backpointer path) is known and
- path discarding will be performed only for nodes in OPEN, no node in CLOSED will be reopened.

When using a monotone heuristic function in A^*_ϵ , this is not true in general.

ϵ -Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in A^*_ϵ

When using a monotone heuristic function in A^* ,

- at time of expansion of a node n an optimal path from s to n (the backpointer path) is known and
- path discarding will be performed only for nodes in OPEN, no node in CLOSED will be reopened.

When using a monotone heuristic function in A^*_ϵ , this is not true in general.

→ Restricted Parent Discarding

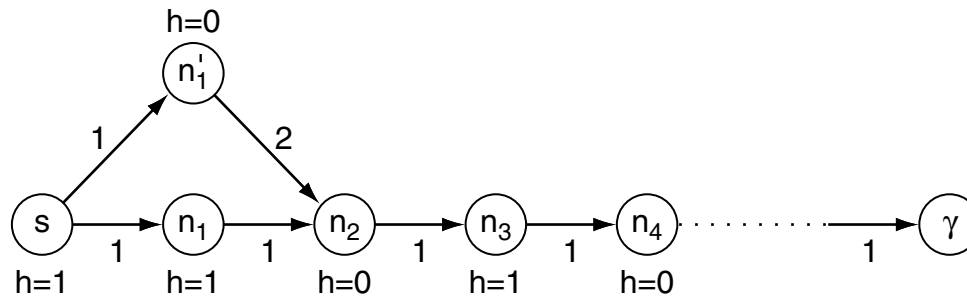
Parent discarding is applied only for nodes in OPEN, i.e. only for nodes that have not been expanded.

An A^*_ϵ algorithm using restricted path discarding is called NRA^*_ϵ .

What are the consequences of using restricted path discarding with respect to ϵ -admissibility?

ϵ -Admissible Speedup Versions of A*

Example: Monotone Heuristic Function h in A^*_ϵ



Let $s, n_1, n_2, \dots, \gamma$ be an optimal solution path and $\epsilon = \frac{1}{2}$.

A^*_ϵ uses heuristic function $h_F = h$.

- Node n_2 is suboptimally reached, but nevertheless expanded.
 - Then n_1 is expanded and – due to path discarding – n_2 will be reopened.
- Reopening cannot be avoided in A^*_ϵ although a monotone heuristic function h is used.

ε -Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in A^*_ε

Lemma 73 (ε -Restricted Reopening)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. When using a monotone heuristic function h in algorithm A^*_ε the deviation of the cost of the backpointer path of an expanded node from its optimal path cost is limited, i.e., for any node n in CLOSED we have

$$g(n) - g^*(n) \leq \varepsilon \cdot (g^*(n) + h(n))$$

ε -Admissible Speedup Versions of A^*

Using Monotone Heuristic Functions h in A^*_ε

Lemma 73 (ε -Restricted Reopening)

Let G be a search space graph with $Prop(G)$ and $\varepsilon > 0$. When using a monotone heuristic function h in algorithm A^*_ε the deviation of the cost of the backpointer path of an expanded node from its optimal path cost is limited, i.e., for any node n in CLOSED we have

$$g(n) - g^*(n) \leq \varepsilon \cdot (g^*(n) + h(n))$$

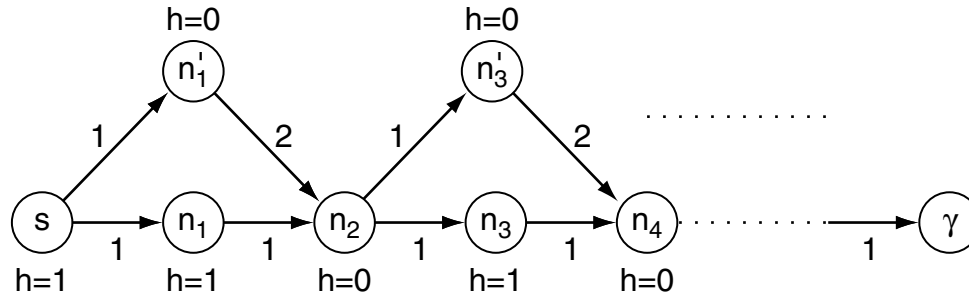
Proof (sketch)

Let s, \dots, n', \dots, n be an optimal path from s to n . At time of expansion of n let n' be the shallowest OPEN node in that path and let n_0 be a node with smallest f -value in OPEN. Then we have

$$\begin{aligned} f(n) &\leq (1 + \varepsilon) \cdot f(n_0) \\ &\leq (1 + \varepsilon) \cdot f(n') \\ &\leq (1 + \varepsilon) \cdot (g^*(n') + h(n')) \leq (1 + \varepsilon) \cdot (g^*(n') + k(n', n) + h(n)) \\ &= (1 + \varepsilon) \cdot (g^*(n) + h(n)) \end{aligned}$$

ϵ -Admissible Speedup Versions of A*

Example: Monotone heuristic function h in NRA^*_ϵ



Let $s, n_1, n_2, \dots, \gamma$ be an optimal solution path, let $\epsilon = \frac{1}{2}$.

NRA^*_ϵ uses heuristic function $h_F = h$.

NRA^*_ϵ uses restricted path discarding.

- Node n_2 is suboptimally reached, but nevertheless expanded.
- Then n_1 is expanded and—due to restricted path discarding— n_2 will not be reopened.
- The deviation to optimal path cost increases with each non-reopening and hence depends on the length of paths.

ε -Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in NRA^*_ε

Theorem 74 (Bounded Admissibility of NRA^*_ε)

Let G be a search space graph with $\text{Prop}(G)$ containing solution paths and let $\varepsilon > 0$. Let N be the maximal length of an optimal solution path. If the heuristic function h is monotone, algorithm NRA^*_ε terminates with solution cost C with

$$C \leq (1 + \varepsilon) \lceil \frac{N}{2} \rceil \cdot C^*$$

ε -Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in NRA^*_ε

Theorem 74 (Bounded Admissibility of NRA^*_ε)

Let G be a search space graph with $\text{Prop}(G)$ containing solution paths and let $\varepsilon > 0$. Let N be the maximal length of an optimal solution path. If the heuristic function h is monotone, algorithm NRA^*_ε terminates with solution cost C with

$$C \leq (1 + \varepsilon)^{\lfloor \frac{N}{2} \rfloor} \cdot C^*$$

Proof (sketch)

- ❑ Consider an optimal solution path. Then the path length is bounded by N .
- ❑ Restricted path discarding occurs on this path if
 - a node that is suboptimally reached is expanded and
 - a predecessor node is expanded later.
- ❑ Analogously to the preceding lemma it can be shown that the deviation in g -values is limited for each occurrence of restricted path discarding.
- ❑ Since two new nodes must always be involved for an increase in deviation of a g -value to occur, the deviation of a g -value from g^* increases at most $\lfloor \frac{N}{2} \rfloor$ times.