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 optimum 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]:

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

- $w = 0 \rightsquigarrow$ 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?

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

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 88 (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).
- ➜ When considering graphs G with *Prop*$_{A^*}(G)$, all results for A*, which do not require further restrictions on the heuristic functions h, also apply to 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).
- ➜ When considering graphs G with *Prop*$_{A^*}(G)$, all results for A*, which do not require further restrictions on the heuristic functions h, also apply to WA*.

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

Remarks:

- Property 8 of *Prop*$_{A^*}(G)$ restricts the heuristic function h in A*:

  For each node n in G a heuristic estimate $h(n)$ of the cheapest path cost from n to Γ is computable and $h(n) \geq 0$. Especially, it holds $h(\gamma) = 0$ for $\gamma \in \Gamma$.

  Obviously, if the restrictions are met by a function h, then they are also met by function $(1+\varepsilon)h$ with $\varepsilon \geq 0$.

- A related approach was described by Harris [Harris 1974]. His *Bandwidth Heuristic Search* algorithms is an A* algorithm using a heuristic function h with

  $$h^*(n) - d \leq h(n) \leq h^*(n) + e$$

  with some constants $d, e \geq 0$ for all nodes n in G.

  Taking into account only the right hand side inequality and using an admissible function h for a graph G with *Prop*$_{A^*}(G)$, this algorithm will – in case a solution exists – return a solution with cost C such that $C \leq C^* + e$.

  However, such a bandwidth restriction for values of the heuristic function can only exist if the condition $h(n) < +\infty \Leftrightarrow h^*(n) < +\infty$ holds. Obviously, there is no need to store a node n with $h(n) = \infty$ on OPEN, since there is no path from n to a goal node in G. Then, the bandwidth condition allows us to drop a node n with $h(n) < +\infty$ from OPEN whenever there is another node n' in OPEN with with $h(n') < +\infty$ such that $f(n') < f(n) - (e + d)$.

  When dropping nodes from OPEN, it is essential to verify that shallowest OPEN nodes of optimum solution paths will never be dropped.

ε-Admissible Speedup Versions of A*

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

Theorem 89 (ε-Admissibility of WA*)

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

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

ε-Admissible Speedup Versions of A*

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

Theorem 89 (ε-Admissibility of WA*)

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

WA* terminates with solution cost C with $C \leq (1+\varepsilon) \cdot C^*$ 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 in *Prop*$_{A^*}(G)$ are also met by $(1+\varepsilon) \cdot h$.
2. Let WA* terminate with goal node γ and solution cost $C = f_\varepsilon(\gamma)$.
3. Let n' be the shallowest OPEN node on some optimum solution path at termination. Then we have $f_\varepsilon(n') = g^*(n') + (1+\varepsilon) \cdot h(n') \leq (1+\varepsilon) \cdot (g^*(n') + h(n'))$.
   [Corollary "Shallowest OPEN Node on Optimum Path" also holds for WA*]
4. Since h is admissible, we have $f_\varepsilon(n') \leq (1+\varepsilon) \cdot (g^*(n') + h^*(n'))$
5. From $g^*(n') + h^*(n') = C^*$ (node on optimum path) follows that $f_\varepsilon(n') \leq (1+\varepsilon) \cdot C^*$.
6. Since WA* selects nodes with smallest f_ε-values, we have $C \leq f_\varepsilon(n') \leq (1+\varepsilon) \cdot C^*$.

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:

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

depth(n): depth of node n (length of back-pointer path to n)

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

- *depth*$(n) \ll N$: h is given a supportive weight equal to $(1+\varepsilon)$.

  ➜ Depth-first excursions are encouraged.

- *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 $\varepsilon \longrightarrow 0$ we have $f_{(d)\varepsilon}(n) \longrightarrow g(n) + h(n)$.
- 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.
- A revised version of DWA* uses a ratio of estimated distances to to goal nodes:

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

  The resulting algorithm is called RDWA* [Thayer & Ruml 2009].

  "If $d(n)$ is an accurate estimate of the length of a cost-optimal path from n to a goal node, then revised dynamically weighted A* will only reward progress towards a goal instead of rewarding all movement away from the root."

ε-Admissible Speedup Versions of A*

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

Theorem 90 (ε-Admissibility of DWA*)

Let G be a search space graph with *Prop*$_{A^*}(G)$ 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 90 (ε-Admissibility of DWA*)

Let G be a search space graph with $\textit{Prop}_{A^*}(G)$ 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

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

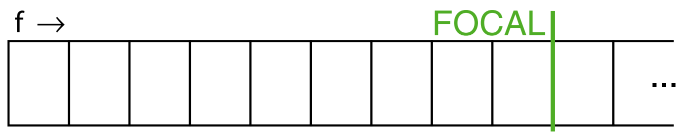
2. Therefore we have $C \leq f_{d\varepsilon}(n') \leq (1 + \varepsilon) \cdot C^*$.

ε-Admissible Speedup Versions of A*

Node Selection by $h_F(n)$: A*$_\varepsilon$ [Pearl/Kim 1982]

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

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



f-sorted OPEN

➜ 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 $\varepsilon > 0$ is called A*$_\varepsilon$.

Remarks:

- Depth of a node in the traversal tree can be seen an indication of computational effort required to solve the rest problem for that node.

- Clearly, for $\varepsilon = 0$, A*$_\varepsilon$ 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*$_\varepsilon$ 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*$_\varepsilon$ [Pearl/Kim 1982]

Theorem 91 (ε-Admissibility of A*$_\varepsilon$)

Let G be a search space graph with *Prop*$_{A^*}(G)$ and $\varepsilon > 0$. Then A*$_\varepsilon$ is ε-admissible when using any h_F to select from FOCAL and an admissible heuristic function h.

ε-Admissible Speedup Versions of A*

Node Selection by $h_F(n)$: A*$_\varepsilon$ [Pearl/Kim 1982]

Theorem 91 (ε-Admissibility of A*$_\varepsilon$)

Let G be a search space graph with *Prop*$_{A^*}(G)$ and $\varepsilon > 0$. Then A*$_\varepsilon$ is ε-admissible when using any h_F to select from FOCAL and an admissible heuristic function h.

Proof (sketch)

1. Completeness of A*$_\varepsilon$ 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*$_\varepsilon$ terminate with goal node γ and solution cost $C = f(\gamma)$.
3. Let n' be the shallowest OPEN node on some optimum solution 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*$_\varepsilon$ 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*$_\varepsilon$.

  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*$_\varepsilon$.

ε-Admissible Speedup Versions of A*

Comparison of DWA* and A*$_\varepsilon$

- 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*$_\varepsilon$:

  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 back-pointer path from s to n, or

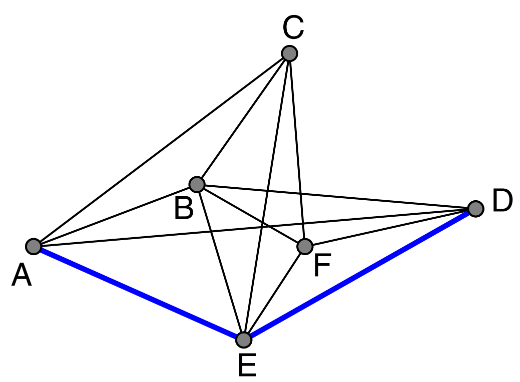
  – utilization of non-additive or non-recursive functions.

ε-Admissible Speedup Versions of A*

Comparison of DWA* and A*$_\varepsilon$ (continued)

Application of A*, DWA* and A*$_\varepsilon$ 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*$_\varepsilon$ use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j, while i ranges over the *unvisited* cities and j ranges over the *all* cities.
- DWA* uses $N = 9$ (search depth is 9), DWA* and A*$_\varepsilon$ use $\varepsilon \in (0; 0.2]$.
- The focal-heuristic h_F of A*$_\varepsilon$ is the number of unvisited cities.

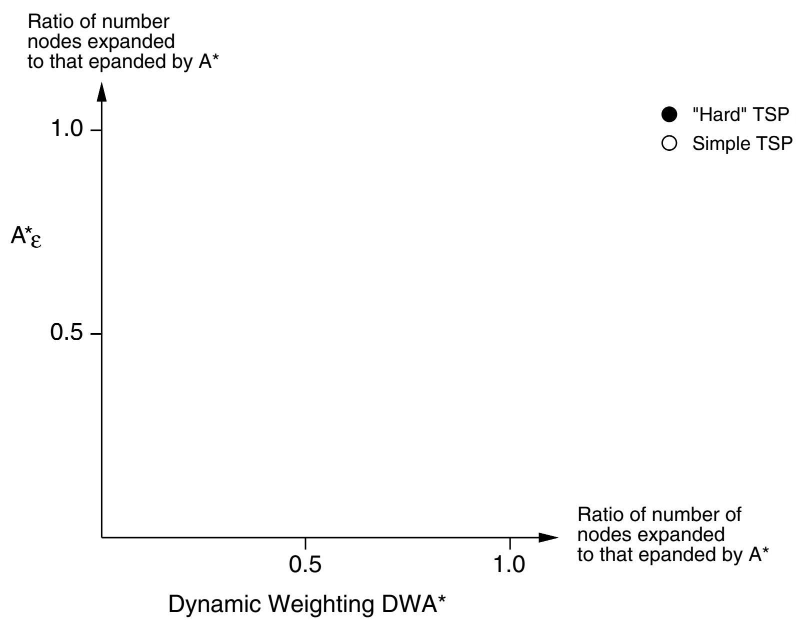


ε-Admissible Speedup Versions of A*

Comparison of DWA* and A*$_\varepsilon$ (continued)

Application of A*, DWA* and A*$_\varepsilon$ 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*$_\varepsilon$ use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j, while i ranges over the *unvisited* cities and j ranges over the *all* cities.
- DWA* uses $N = 9$ (search depth is 9), DWA* and A*$_\varepsilon$ use $\varepsilon \in (0; 0.2]$.
- The focal-heuristic h_F of A*$_\varepsilon$ is the number of unvisited cities.

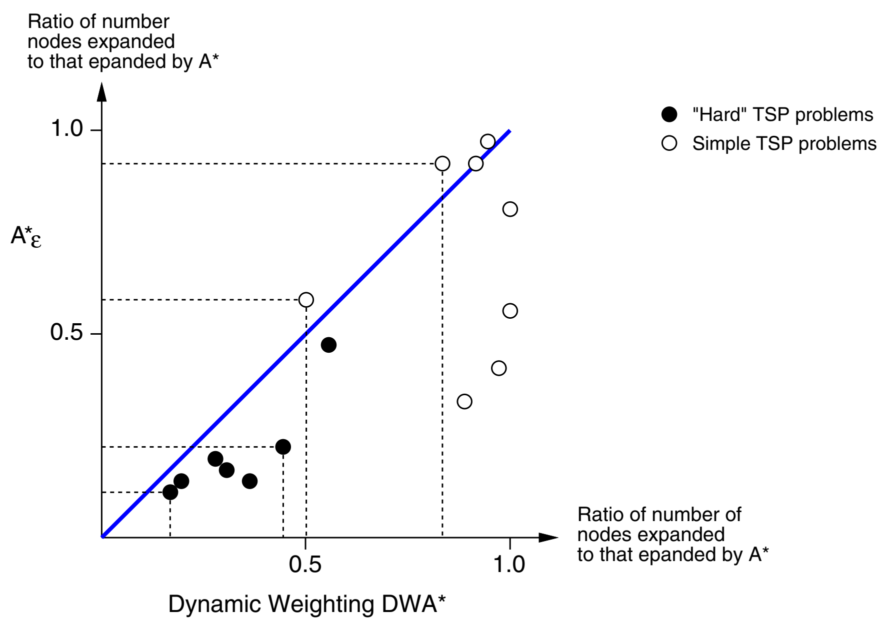


ε-Admissible Speedup Versions of A*

Comparison of DWA* and A*$_\varepsilon$ (continued)

Application of A*, DWA* and A*$_\varepsilon$ 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*$_\varepsilon$ use $h = \sum_i \min_{j \neq i} d_{ij}$, where d_{ij} is the distance between city i and city j, while i ranges over the *unvisited* cities and j ranges over the *all* cities.
- DWA* uses $N = 9$ (search depth is 9), DWA* and A*$_\varepsilon$ use $\varepsilon \in (0; 0.2]$.
- The focal-heuristic h_F of A*$_\varepsilon$ is the number of unvisited cities.



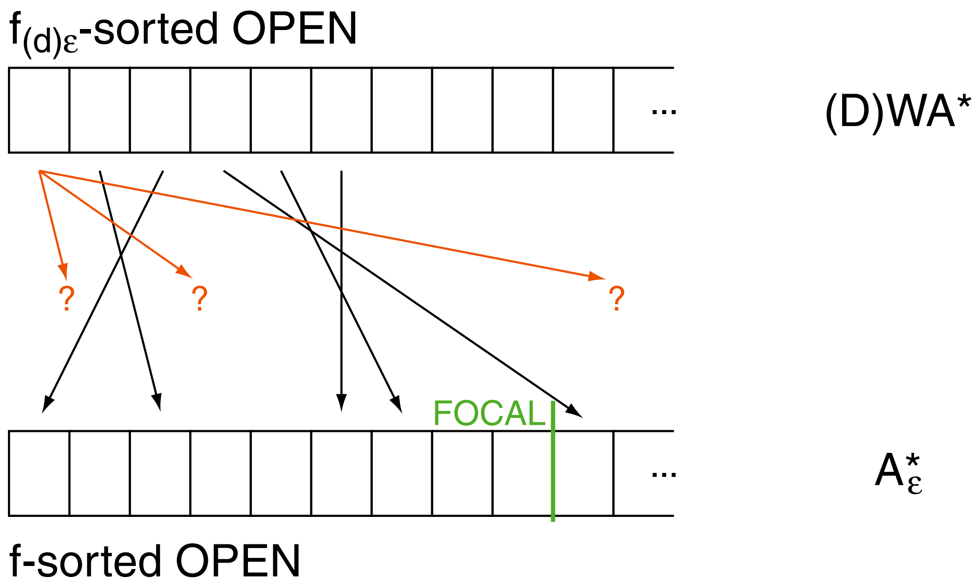
Remarks:

- 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 $\left(1 - \frac{\mathit{depth}(n)}{N}\right)$, which guides the dynamic weighting algorithm.

ε-Admissible Speedup Versions of A*

Unifying View: WA* and DWA* as Variants of A*$_\varepsilon$

Approach: Use $h_F = f_\varepsilon$ resp. $h_F = f_{d\varepsilon}$ in A*$_\varepsilon$.



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

Remarks:

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

ε-Admissible Speedup Versions of A*

Lemma 92 (WA* and DWA* are variants of A*$_\varepsilon$)

Let G be a search space graph with *Prop*$_{A^*}(G)$ 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 $n'_0 := \text{argmin}_{n \in \text{OPEN}} f'(n)$ we have

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

ε-Admissible Speedup Versions of A*

Lemma 92 (WA* and DWA* are variants of A*$_\varepsilon$)

Let G be a search space graph with *Prop*$_{A^*}(G)$ 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 $n'_0 := \text{argmin}_{n \in \text{OPEN}} f'(n)$ we have

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

Proof (sketch)

Let $n_0 := \text{argmin}_{n \in \text{OPEN}} f(n)$. 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*$_\varepsilon$ [A* Condition II]

Corollary 93 (Necessary Condition for Node Expansion II for A*$_\varepsilon$)

Let G be a search space graph with *Prop*$_{A^*}(G)$, an admissible heuristic function h, and $\varepsilon > 0$. For any node n expanded by A*$_\varepsilon$ 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*$_\varepsilon$).
- 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*$_\varepsilon$

When using a monotone heuristic function in A*,

- at time of expansion of a node n an optimal path from s to n (the back-pointer 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*$_\varepsilon$, this is not true in general.

ε-Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in A*$_\varepsilon$

When using a monotone heuristic function in A*,

- at time of expansion of a node n an optimal path from s to n (the back-pointer 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*$_\varepsilon$, this is not true in general.

➜ Restricted Parent Discarding

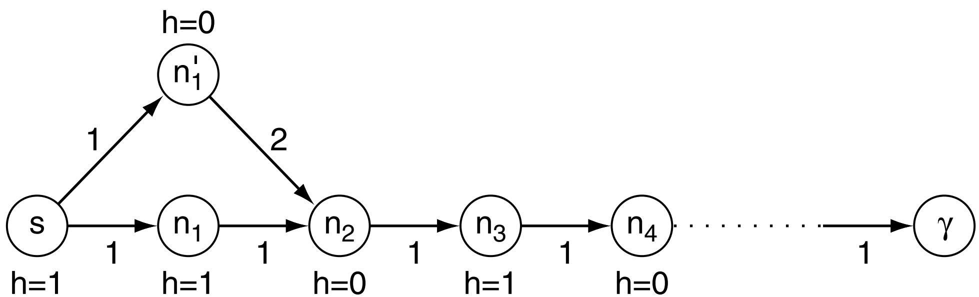
Parent discarding is applied only for nodes in OPEN, i.e. only for nodes that have not been expanded.

An A*$_\varepsilon$ algorithm using restricted path discarding is called NRA*$_\varepsilon$.

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*$_\varepsilon$



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

A*$_\varepsilon$ 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*$_\varepsilon$ although a monotone heuristic function h is used.

ε-Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in A*$_\varepsilon$ (continued)

Lemma 94 (ε-Restricted Reopening)

Let G be a search space graph with *Prop*$_{A^*}(G)$ and $\varepsilon > 0$. When using a monotone heuristic function h in algorithm A*$_\varepsilon$ the deviation of the cost of the back-pointer 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*$_\varepsilon$ (continued)

Lemma 94 (ε-Restricted Reopening)

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

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

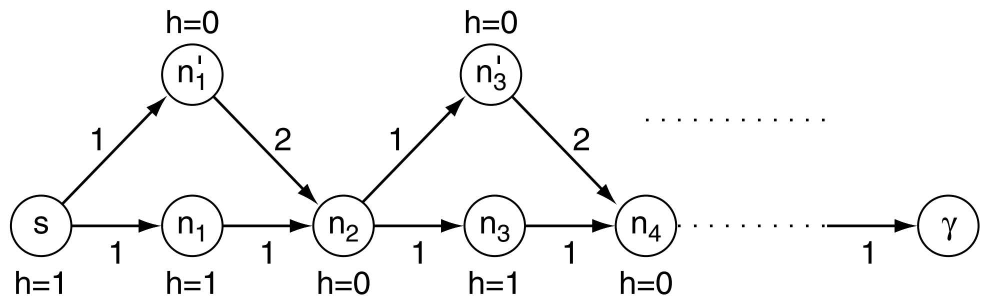
Proof (sketch)

Let $s, \ldots, n', \ldots, n$ 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. The 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*$_\varepsilon$



Let $s, n_1, n_2, \ldots, \gamma$ be an optimum solution path, let $\varepsilon = \frac{1}{2}$.

NRA*$_\varepsilon$ uses heuristic function $h_F = h$.

NRA*$_\varepsilon$ 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*$_\varepsilon$

Theorem 95 (Bounded Admissibility of NRA*$_\varepsilon$)

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

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

ε-Admissible Speedup Versions of A*

Using Monotone Heuristic Functions h in NRA*$_\varepsilon$

Theorem 95 (Bounded Admissibility of NRA*$_\varepsilon$)

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

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

Proof (sketch)

- Consider an optimum 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.