On the Optimal Efficiency of Cost-Algebraic A*

Robert C. Holte
Computing Science Dept.
University of Alberta
Edmonton, Canada T6G 2E8
(rholte@ualberta.ca)

Sandra Zilles
Computer Science Dept.
University of Regina
Regina, Canada S4S 0A2
(zilles@cs.uregina.ca)

Abstract

Edelkamp et al. (2005) proved that A*, given an admissible heuristic, is guaranteed to return an optimal solution in any cost algebra, not just in the traditional shortest path setting. In this paper, we investigate cost-algebraic A*'s optimal efficiency: in the cost-algebraic setting, under what conditions is A* guaranteed to expand the fewest possible states? In the traditional setting, this question was examined in detail by Dechter & Pearl (1985). They identified five different situations in which A* was optimally efficient. We show that three of them continue to hold in the cost-algebraic setting, but that one does not. We also show that one of them is false, it does not hold even in the traditional setting. We introduce an alternative that does hold in the cost-algebraic setting. Finally, we show that a well-known result due to Nilsson does not hold in the general cost-algebraic setting but does hold in a slightly less general setting.

1 Introduction

In the traditional shortest path problem, the cost of an edge can be any non-negative real number, the cost of a path is the sum of its edge costs, and the aim is to find a path from the start state to a goal state whose cost is smallest according to the usual way of ordering the real numbers. This particular combination of permissible values, cost-aggregating operator, and value ordering is an example of a cost algebra. There are many other ways of defining the set of permissible values, cost-aggregating operator, and value ordering, and some of these alternatives are of practical importance, in network routing for example (Sobrinho 2002).

An example of a "shortest path" problem based on a different cost algebra is the widest path problem. A robot travels to a goal location through a network of corridors, each of which has a width. The aim is to find a path whose narrowest corridor is widest, so as to minimize the risk of the robot scraping against a wall. The values in this problem are the positive real values up to some maximum possible width, the cost-aggregating operator is $\min$ (because it is the narrowest corridor in a path that determines the path's value), and the value ordering is the opposite of the ordering used for the normal shortest path problem (because this is a maximization problem, not a minimization problem).

A* (Hart, Nilsson, and Raphael 1968) is a fundamental algorithm for solving the traditional shortest path problem, so the question naturally arises, can it also be used to solve shortest path problems in the general cost-algebraic setting (e.g. the widest path problem)? Edelkamp et al. (2005) showed that the answer is yes. Just as in the traditional setting, A* with an admissible heuristic is guaranteed to return an optimal solution in the general cost-algebraic setting.

In this paper our primary focus is on the optimal efficiency of A*: under what conditions is A* guaranteed to expand the fewest possible states?

In the traditional setting, this question was examined in detail by Dechter & Pearl (1985). They identified five different situations in which A* was optimally efficient ("0-optimal" in their terminology), which we refer to as results R1 through R5 (see Section 5 for details). We show that R1, R2, and R5 continue to hold in the cost-algebraic setting, but that R4 does not. We also show that R3 is false, it does not hold even in the traditional setting. We introduce an alternative to R3 that does hold in the cost-algebraic setting.

Finally, we consider the following result in the traditional setting (Nilsson 1980): if heuristics h_1 and h_2 are admissible and $h_1(s) > h_2(s)$ on every non-goal state s then every state expanded by A* using h_1 is also expanded by A* using h_2. We show that this does not hold in the general cost-algebraic setting but does hold in a slightly less general setting.

Proofs of some lemmas are omitted due to space limitations. A technical report with all the proofs is available from the authors on request.

2 Cost Algebra Definitions

The definitions in this section are copied from (Edelkamp, Jabbar, and Lluch Lafuente 2005) with a few minor changes.

Informally, Ω is the set of possible edge and path costs, $\mathbf{1}$ is the cost of the empty path, $\times$ is the operator for computing a path's cost from the costs of its edges, and $\preceq$ is the ordering on costs. Those four components define a cost algebra. Isotonicity is a key defining property of a cost algebra, almost all our theoretical results rely on it.

Definition 1. *Let Ω be a set and $\times : \Omega \times \Omega \to \Omega$ a binary operator. A monoid is a triple $\langle \Omega, \times, \mathbf{1} \rangle$ such that $\mathbf{1} \in \Omega$ and for all $a, b, c \in \Omega$:*

- $a \times (b \times c) = (a \times b) \times c$

- $\mathbf{1} \times a = a \times \mathbf{1} = a$ *($\mathbf{1}$ is the identity element)*

Definition 2. *If $\langle \Omega, \times, \mathbf{1} \rangle$ is a monoid and $\langle a_1, \ldots, a_n \rangle \in \Omega^n$ is a sequence of length $n \geq 0$, then*

$$\prod_{i=1}^{n} a_i = \begin{cases} \mathbf{1}, & \textit{if } n = 0 \\ a_1 \times a_2 \times \cdots \times a_n, & \textit{otherwise.} \end{cases}$$

Definition 3. *If Ω is a set and $\preceq$ is a total order on Ω then $a \prec b$ denotes $(a \preceq b) \wedge (a \neq b)$, and $a \succeq b$ and $a \succ b$ are alternative ways of writing $b \preceq a$ and $b \prec a$ respectively.*

Definition 4. *A monoid $\langle \Omega, \times, \mathbf{1} \rangle$ with total order $\preceq$ on Ω is* isotone *iff $a \preceq b$ implies both $(a \times c) \preceq (b \times c)$ and $(c \times a) \preceq (c \times b)$ for all $a, b, c \in \Omega$.*

In cost-algebraic search, where the elements of Ω are path costs, isotonicity is a very natural property. For two paths P_a and P_b costing a and b, respectively, with $a \preceq b$, isotonicity requires that appending a path of cost c to P_a produces a path whose total cost is less than or equal to ($\preceq$) the total cost of the path produced by appending a path of cost c to P_b. This is closely related to the "order preserving" property defined by Dechter & Pearl.

Definition 5. *A cost algebra is a 4-tuple $\langle \Omega, \times, \preceq, \mathbf{1} \rangle$ such that*

- *$\langle \Omega, \times, \mathbf{1} \rangle$ is a monoid*
- *$\preceq$ is a total order on Ω*
- *$\mathbf{1} \preceq a$ for all $a \in \Omega$*
- *$\langle \Omega, \times, \mathbf{1} \rangle$ with $\preceq$ is isotone.*

3 Search-Related Definitions

Many A* studies, including Dechter & Pearl's, allow a state space to contain an infinite number of states but require edge costs to be bounded away from zero. This requirement is used to guarantee that A* terminates after a finite number of iterations but does not play a role in Dechter & Pearl's optimal efficiency results. We take an alternative approach, allowing edge costs to be 0 ($\mathbf{1}$ in the algebra) but restricting the state space to be finite in order to guarantee A*'s termination.

For a finite set of states S and cost algebra $\langle \Omega, \times, \preceq, \mathbf{1} \rangle$, a problem $\mathcal{P}$ is a triple $\langle start, goal, \sigma \rangle$, where $start, goal \in S$ and σ is a successor function on S.[1] A successor function maps each $s \in S$ to a set of pairs $\{(s_i, \omega(s, s_i))\}$ where $s_i \in S$ and $\omega : S \times S \rightarrow \Omega$ is a cost function specifying the cost, $\omega(s, t)$, of transitioning directly from state s to state t. If $\sigma(s)$ contains a pair of the form (t, x), we write $t \in \sigma(s)$, t is called a successor of s, and we say there is an edge from s to t whose cost is x. To identify the problem $\mathcal{P}$ with which a successor function is associated we will use $\mathcal{P}$ as a subscript, *i.e.* $\sigma_{\mathcal{P}}$. Σ denotes the set of all possible successor functions for a given state set and cost algebra.

For a given $\sigma \in \Sigma$, a finite sequence $\langle s_0, \ldots, s_n \rangle$ of states in S is a path if $s_i \in \sigma(s_{i-1})$ for $i \in [1, n]$. If $P = \langle s_0, \ldots, s_n \rangle$ is a path then its cost, $\omega(P)$, is $\prod_{i=1}^{n} \omega(s_{i-1}, s_i)$. Path P from state s to state t is optimal if $\omega(P) \preceq \omega(P')$ for all paths P' from s to t. If there exists a path from state s to state t the distance from s to t, $d(s, t)$, is the cost of an optimal path from s to t. If there is no path from s to t then $d(s, t) = \infty$, where ∞ is a special value ($\infty \notin \Omega$) and $\preceq$ and $\times$ are extended as follows:

- $a \prec \infty$ for all $a \in \Omega$
- $a \times \infty = \infty \times a = \infty \times \infty = \infty$ for all $a \in \Omega$.

When we wish to emphasize the specific σ on which distances (and related functions such as ω) depend we will add σ as a subscript, as in $d_\sigma(s, t), \omega_\sigma(P)$, etc.

Problem $\langle start, goal, \sigma \rangle$ is solvable if there exists a path from $start$ to $goal$. In this paper we are exclusively concerned with solvable problems, so we will henceforth use the term "problem" to mean "solvable problem". A solution for problem $\mathcal{P}$ is a path from $start$ to $goal$, and $C^*_{\mathcal{P}} = d(start, goal)$ is the cost of an optimal solution (written as C^* when $\mathcal{P}$ is obvious from context).

A search algorithm succeeds on problem $\mathcal{P}$ if it returns $C^*_{\mathcal{P}}$ when given $\mathcal{P}$ as input, and succeeds on a set of problems $\mathbb{P}$ if it succeeds on every problem in $\mathbb{P}$.

A heuristic is a function $h : S \rightarrow \Omega \cup \{\infty\}$ with $h(goal) = \mathbf{1}$. We will write A^*_h to denote A* using h as its heuristic.

Definition 6. *For heuristic h, $goal \in S$, and $\sigma \in \Sigma$, the triple $\langle h, goal, \sigma \rangle$ is* admissible *iff $h(s) \preceq d_\sigma(s, goal)$ for all $s \in S$ and is* consistent[2] *iff $h(s) \preceq \omega_\sigma(s, t) \times h(t)$ for all $s \in S$ and $t \in \sigma(s)$.*

Definition 7. *For heuristic h, $\mathbb{P}_{AD}(h)$ is the set of problems $\langle start, goal, \sigma \rangle$ such that $\langle h, goal, \sigma \rangle$ is admissible, and $\mathbb{P}_{CONS}(h)$ is the set of problems $\langle start, goal, \sigma \rangle$ such that $\langle h, goal, \sigma \rangle$ is consistent.*

$\mathbb{P}_{AD}(h)$ and $\mathbb{P}_{CONS}(h)$ correspond to Dechter & Pearl's I_{AD} and I_{CONS}, respectively.

Definition 8. *Let h be any heuristic and $\mathcal{P}$ a problem. Then path $P = \langle s_0, \ldots, s_n \rangle$ is* strictly C^*-bounded *iff $f_P(s_i) \prec C^*$ for all $i \in [0, n]$, where $f_P(s_i) = g_P(s_i) \times h(s_i)$, and*

$$g_P(s_i) = \prod_{j=0}^{i-1} \omega(s_j, s_{j+1}).$$

State s is surely expanded by A* *(abbreviated s.e.) iff there is a path from start to s that is strictly C^*-bounded.*

Definition 9. *Given any heuristic h, problem $\langle start, goal, \sigma \rangle$ is non-pathological iff there exists an optimal solution $P = \langle s_0, \ldots, s_n \rangle$ ($s_0 = start, s_n = goal$) such that $f_P(s_i) \prec C^*$ for $i \in [0, n-1]$.*

[1] This definition mandates that a problem have just one goal state. Dechter & Pearl allow multiple goal states, but this difference does not materially affect any of our analysis.

[2] Historically, this is the definition of a heuristic being "monotone", consistency required $h(s) \preceq d_\sigma(s, t) \times h(t)$ for all $s, t \in S$. Pearl (1984) was the first to note that the two definitions are equivalent in the traditional numerical, additive setting. This equivalence also holds in the cost-algebraic setting.

4 Fundamental A* Results

In this section we prove that A*, in the cost-algebraic setting, always expands all s.e. states (Lemma 2) and, if the problem being solved is non-pathological, it expands only the s.e. states (Lemma 3). Both results use the following generalization of Lemma 1 by Hart et al. (1968). Our proof closely follows theirs.

Lemma 1. *The following holds at the beginning of every iteration prior to A*'s termination: if state s is not closed, then for every path P from start to s there exists a state $s' \in P$ that is in Open with $g(s') \preceq g_P(s')$.*

Proof: There are two cases to consider.
Case 1: $start$ is not closed. In this case the lemma follows with $start$ playing the role of s' for all states s and paths P from $start$ to s.
Case 2: $start$ is closed. Let s be any state that is not closed, P any path from $start$ to s, and Δ the set of all states $s_i \in P$ that are closed with $g(s_i) \preceq g_P(s_i)$. Δ is not empty because $start \in \Delta$. Let s^* be the state in Δ with the largest "index" in P, i.e. all other states in Δ precede s^* in P. Let t be the successor of s^* in P (t exists because $s^* \neq s$ since the latter is not in Δ).

We will now show that t has the properties required of s' by the lemma. At the time s^* became closed with its present g-value, $g(s^*)$, a path to t was created with a cost of $g(s^*) \times \omega(s^*, t)$. Because of isotonicity and the fact that $g(s^*) \preceq g_P(s^*)$, we have $g(s^*) \times \omega(s^*, t) \preceq g_P(s^*) \times \omega(s^*, t) = g_P(t)$. The current g-value of t, $g(t)$, cannot be larger than $g(s^*) \times \omega(s^*, t)$ (because g-values can only decrease as A* proceeds) and therefore $g(t) \preceq g_P(t)$.

Because t has been generated (by the expansion of s^*) it is either open or closed. We will now show it cannot be closed, and therefore must be open. t cannot be closed with $g(t) \preceq g_P(t)$ at the present time because if it were, it would be in Δ, contradicting the fact that s^* has the largest index of the states in Δ. t could have been in $Closed$ with $g(t) \succ g_P(t)$ prior to s^* being closed with its present g-value, but A* would have removed t from $Closed$ at that time because a lower cost path to t was generated. Therefore at the present time, t must be open with $g(t) \preceq g_P(t)$, so $s' = t$ establishes the lemma. □

Lemma 2. *Let h be a heuristic and $\mathcal{P}$ a problem. If state s is s.e. then A^*_h will expand it.*

Proof: Because s is s.e. there exists a strictly C^*-bounded path P from $start$ to s, i.e. a path such that $f_P(s_i) \prec C^*$ for all $s_i \in P$. By Lemma 1, if s is not closed then there exists an open state $s' \in P$ with $g(s') \preceq g_P(s')$. Because of isotonicity, this implies $g(s') \times h(s') \preceq g_P(s') \times h(s')$, i.e. that $f(s') \preceq f_P(s')$. We have $f_P(s') \prec C^*$ and, if $goal$ is on $Open$, $C^* \preceq f(goal)$, so that $f_P(s') \prec f(goal)$. Since A^*_h terminates when $f(goal)$ is the smallest ($\preceq$) f-value among the states in $Open$, it will not terminate until all the states on P, including s, have been expanded. □

Lemma 3. *Let h be a heuristic and $\mathcal{P}$ a non-pathological problem. If A^*_h breaks ties in favour of goal then it will return an optimal solution for $\mathcal{P}$ and will only expand s.e. states.*

5 Dechter & Pearl's Optimal Efficiency Results

Dechter & Pearl (1985) give four definitions of "optimality". Of the most practical interest is what they called 0-optimal:

Definition 10. *Let $\mathbb{P}$ be a set of problems and $\mathbb{A}$ a set of algorithms. Then A* is* 0-optimal *over $\mathbb{A}$ relative to $\mathbb{P}$ iff for every $\mathcal{P} \in \mathbb{P}$ and $\mathcal{A} \in \mathbb{A}$ the set of nodes expanded by A* in solving $\mathcal{P}$, no matter what tie-breaking rule A* uses as long as ties are broken in favour of goal, is a subset of the set of nodes expanded by $\mathcal{A}$ in solving $\mathcal{P}$.*

We will focus on the situations in which Dechter & Pearl proved that A* is 0-optimal and use the phrase "optimally efficient" to mean 0-optimal.

Dechter & Pearl analyzed 12 different situations: problems could be drawn from $\mathbb{P}_{AD}(h)$ or $\mathbb{P}_{CONS}(h)$, problems could be pathological or non-pathological, and the algorithms to which A* is compared could be one of three types: ad (admissible, i.e. guaranteed to succeed on $\mathbb{P}_{AD}(h)$), *gc* (defined in Section 6 below), or *bf* (what we call BF*(F) in Section 7 below). The following are all the combinations of problem- and algorithm-types they analyzed for which A* is 0-optimal (see their Figure 9):

(R1) Non-pathological $\mathcal{P} \in \mathbb{P}_{CONS}(h)$, ad algorithms (their Theorem 8). Their proof, with only minor changes in a few details, is valid in the cost-algebraic setting. We present this in Appendix A.

(R2) Non-pathological $\mathcal{P} \in \mathbb{P}_{CONS}(h)$, *gc* algorithms. Follows from R1 because every *gc* algorithm is ad.

(R3) Non-pathological $\mathcal{P} \in \mathbb{P}_{AD}(h)$, *gc* algorithms (their Corollary 5). In Section 6 we show that this result is false, even in the traditional setting. In Section 6.1 we give an alternative result that maintains some of the spirit of R3.

(R4) Non-pathological $\mathcal{P} \in \mathbb{P}_{AD}(h)$, *bf* algorithms (their Corollary 7). In Section 7 we give an example showing that this result does not hold, in general, in the cost-algebraic setting.

(R5) Non-pathological $\mathcal{P} \in \mathbb{P}_{CONS}(h)$, *bf* algorithms. For Dechter & Pearl this is a special case of R4, so they did not give a separate proof. However, their proof of R4, with only minor changes in a few details, is valid in the cost-algebraic setting when restricted to $\mathbb{P}_{CONS}(h)$, as we will prove in Section 7.

6 Result R3 (Dechter & Pearl's Corollary 5)

Dechter & Pearl's Corollary 5 deals with an important type of situation—an admissible, but not necessarily consistent, heuristic is available for the given problem—but only considers a particular type of search algorithm, called a globally compatible (*gc*) algorithm. They define an algorithm to be *gc* if it returns an optimal solution to a problem whenever A* does, even if the heuristic used by A* is inadmissible.

In our notation, Dechter & Pearl's Corollary 5 is this:

> For any heuristic h, non-pathological problem $\mathcal{P} \in \mathbb{P}_{AD}(h)$, and *gc* algorithm A, if A^*_h breaks ties in favour of $goal$ then any state expanded by A^*_h in solving $\mathcal{P}$ must also be expanded by A in solving $\mathcal{P}$.

Since A^*_h breaking ties in favour of *goal* will only expand s.e. states on a non-pathological problem (Lemma 3), to show that their Corollary 5 is false we need to give a *gc* algorithm and a non-pathological problem with an admissible heuristic that the *gc* algorithm solves optimally without expanding some s.e. state. The proof that our algorithm is *gc* uses the following quantities.

Definition 11. *For path P originating at start define* $Q_f(P) = \max\{f_P(s) | s \in P\}$.

For any state s, define $Q(s) = \min\{Q_f(P) | P$ is a path from start to $s\}$ and $Q_{opt}(s) = \min\{Q_f(P) | P$ is an optimal *path from start to $s\}$.*

Finally, define $Q = Q(goal)$ and $Q_{opt} = Q_{opt}(goal)$.

Our *gc* algorithm, GC1, proceeds in two phases. In Phase I, it computes all least-cost paths from $start$ to a selected state s and updates $h(s)$. In Phase II it runs Algorithm C by Bagchi & Mahanti (1983) – which we will refer to as $AlgC$ – on the given problem but using the updated $h(s)$ value.

Phase I can be done in various ways, for example as follows. Let s be the terminal state of a walk of a predetermined length expanding the leftmost child at each level, and let C_r the cost of that path to s. Then run a depth-first search, without using h, with C_r as the cost bound, recording all the optimal paths to s. The new $h(s)$ value is then $Q_{opt}(s) - d(start, s)$.[3] Thus ends Phase I.

The new $h(s)$ value defined in this way has the important property that it does not change $Q_f(P)$ for any optimal path P from $start$ to s. This property is instrumental in the proof of the following lemma.

Lemma 4. *Given a problem for which $Q = Q_{opt}$, the modified version of the problem produced by GC1's Phase I also has $Q = Q_{opt}$.*

$AlgC$ is identical to A* except in how it chooses nodes for expansion. It records the largest f-value expanded so far in a variable called F. When it selects a node to be expanded, it first looks to see if there are any open nodes n with $f(n) \leq F$. If there are, it chooses one with the smallest g-value. If there are not, it chooses an open node with the smallest f-value, breaking ties in favour of nodes with the smallest g-value. In either case, ties are broken in favour of the goal.

The proof that GC1 is globally compatible with A*, i.e. it returns an optimal solution whenever A* does, is based on two properties of $AlgC$ proven by Bagchi & Mahanti:[4]

(1) the solution returned by A* is never cheaper than the solution returned by $AlgC$ (their Theorem 3.7 and Remark (i) following it). In particular, for our purposes, if A* returns an optimal solution for a problem then so does $AlgC$;

(2) the solution returned by $AlgC$ is optimal if and only if $Q = Q_{opt}$ (their Theorem 3.8).

Together these imply that if A* solves a problem optimally then $Q = Q_{opt}$. The modification of $h(s)$ made in GC1's Phase I does not change this fact (Lemma 4). Since $Q = Q_{opt}$ in the modified problem, (2) guarantees $AlgC$ will return an optimal solution for the modified problem, which, of course, is also an optimal solution for the original problem since GC1 has only modified $h(s)$, not any edge costs. We have thus proven that GC1 returns an optimal solution whenever A* does.

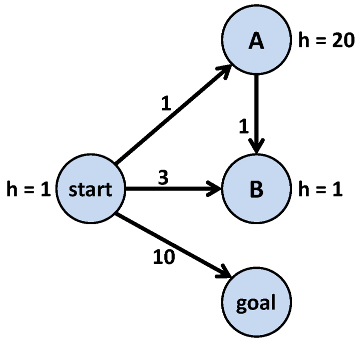

Figure 1: GC1 does not expand s.e. node B.

Figure 1 gives an example of a problem with an admissible heuristic that GC1 solves optimally without expanding s.e. state B. The execution of GC1 on this problem if B is the state s selected in Phase I is as follows. $start$–A–B is the only optimal path from $start$ to B, so $Q_{opt}(B) = Q_f(start$–A–$B) = f(A) = 21$. Phase I therefore updates $h(B)$ to be $21 - 2 = 19$. In Phase II, $AlgC$ halts immediately after expanding $start$ because neither $f(A) = 21$ nor $f(B) = 22$ is less than $f(goal)$. GC1 has found the optimal solution ($start$–$goal$) without expanding B, which is s.e. given its original $h(B)$ value. This completes our counterexample to Dechter & Pearl's Corollary 5.

The key idea in this counterexample is that a state can be s.e. because its initially given heuristic value is excessively low and a process such as GC1's Phase I can increase its heuristic value to the point that the state is no longer s.e.

6.1 An Alternative to Dechter & Pearl's Corollary 5

Definition 12. *Search algorithm A* cost-dominates *search algorithm B iff for any problem $\mathcal{P}$ the cost of the solution returned by A, $C_A(\mathcal{P})$, is no worse than $C_B(\mathcal{P})$, the cost of the solution returned by B (formally, $C_A(\mathcal{P}) \preceq C_B(\mathcal{P})$).*

For example, by proving that the solution returned by A* is never cheaper than the solution returned by $AlgC$, Bagchi & Mahanti proved that $AlgC$ cost-dominates A*.

We will now show that any algorithm that cost-dominates A* must expand all s.e. states. The proof is almost identical to Dechter & Pearl's proof of their Theorem 8 (presented in this paper as Theorem 16 in Appendix A). The key idea in the proof is the construction of a new problem from a given problem $\mathcal{P}$ and s.e. state s. We call the new problem $\mathcal{P}^s_{cd}$. It is identical to $\mathcal{P}$ except it has a new edge directly from s to $goal$ with a suitably chosen cost (c_s). Formally, $\mathcal{P}^s_{cd}$ is defined as follows.

Definition 13. *Given heuristic h, problem $\mathcal{P} = \langle start, goal, \sigma_{\mathcal{P}} \rangle$, and s.e. state s, define $c_s = \mathbf{1}$, and define*

[3] This cannot be smaller than the original $h(s)$ value because s is a state on all optimal paths from $start$ to s.

[4] If Property (2) could be proven for A* then A* could be used in Phase II instead of $AlgC$.

a new problem $\mathcal{P}^s_{cd} = \langle start, goal, \sigma_{\mathcal{P}^s}\rangle$ that is identical to $\mathcal{P}$ except for $\sigma_{\mathcal{P}^s_{cd}}(s)$, which differs from $\sigma_{\mathcal{P}}(s)$ as follows:

- *if $goal \notin \sigma_{\mathcal{P}}(s)$ then $\sigma_{\mathcal{P}^s_{cd}}(s) = \sigma_{\mathcal{P}}(s) \cup \{(goal, c_s)\}$,*
- *if $(goal, \omega_{\mathcal{P}}(s, goal)) \in \sigma_{\mathcal{P}}(s)$ then $\sigma_{\mathcal{P}^s_{cd}}(s) = (\sigma_{\mathcal{P}}(s) \setminus \{(goal, \omega_{\mathcal{P}}(s, goal))\}) \cup \{(goal, c_s)\}$*

Lemma 5. *Let $h, \mathcal{P}, s$, c_s, and $\mathcal{P}^s_{cd}$ be as in Definition 13. Then $C_{A^*}(\mathcal{P}^s_{cd}) \prec C^*_{\mathcal{P}}$.*

A crucial step in Dechter & Pearl's proof of their Theorem 8, which is also needed here, is the assertion that an algorithm that did not expand s.e. state s in solving $\mathcal{P}$ would behave identically on problem $\mathcal{P}$ and the new problem derived from it ($\mathcal{P}^s_{cd}$ in our case). Eckerle et al. (2017) pointed out this is only true of certain kinds of algorithms and identified one set of assumptions that make this assertion true: that the algorithm be deterministic, expansion-based, and "black box", or DXBB (see Appendix B). The following result about DXBB algorithms is what is needed in the proof of Dechter & Pearl's Theorem 8 and in the proof we are about to give about algorithms that cost-dominate A*.

Lemma 6. *Let A be a DXBB algorithm, $\mathcal{P}_1 = \langle start, goal, \sigma_1\rangle$ and $\mathcal{P}_2 = \langle start, goal, \sigma_2\rangle$ two problems with a common start and goal, and X the set of states expanded by A in solving $\mathcal{P}_1$. If $\sigma_1(s) = \sigma_2(s)$ for all $s \in X$, then A will return the same solution when it solves $\mathcal{P}_2$ as it did when it solved $\mathcal{P}_1$.*

Theorem 7. *Let h be a heuristic, $\mathcal{P} = \langle start, goal, \sigma_{\mathcal{P}}\rangle$ a problem, and s an s.e. state. Then any DXBB algorithm A that cost-dominates A* must expand s in solving $\mathcal{P}$.*

Proof: Suppose, for the purpose of contradiction, that A succeeds on problem $\mathcal{P}$ without expanding s.e. state s and let X be the set of states expanded by A in solving $\mathcal{P}$. Since $s \notin X$ and s is the only state on which $\sigma_{\mathcal{P}^s_{cd}}$ and $\sigma_{\mathcal{P}}$ differ, the premises of Lemma 6 are satisfied, so A will return the same solution for problem $\mathcal{P}^s_{cd}$ as it did for $\mathcal{P}$. By Lemma 5, $C_{A^*}(\mathcal{P}^s_{cd}) \prec C^*_{\mathcal{P}}$ and, by definition, $C^*_{\mathcal{P}} \preceq C_A(\mathcal{P}) = C_A(\mathcal{P}^s_{cd})$. Thus, $C_{A^*}(\mathcal{P}^s_{cd}) \prec C_A(\mathcal{P}^s_{cd})$, contradicting the theorem's premise that A cost-dominates A*. □

Combining Theorem 7 and Lemma 3, we get the following optimal efficiency result, which is very much in the spirit of Dechter & Pearl's Theorem 11:

Theorem 8. *For any heuristic h, non-pathological problem $\mathcal{P}$, and DXBB algorithm A that cost-dominates A*, if A^*_h breaks ties in favour of goal then any state expanded by A^*_h in solving $\mathcal{P}$ must also be expanded by A in solving $\mathcal{P}$.*

It is instructive to consider why this theorem, with GC1 serving as algorithm A, is not contradicted by the example in Figure 1. The reason is that GC1 does not cost-dominate A* so Theorem 8 does not apply to GC1. To see that GC1 does not cost-dominate A*, consider the problem $\mathcal{P}^s_{cd}$ created by applying the construction in Definition 13 to the problem in Figure 1 with $s = B$. This $\mathcal{P}^s_{cd}$ is identical to Figure 1 except it has an edge of cost 0 (the algebra's **1**) from B to $goal$. The solution returned by GC1 on this problem would be $start$–$goal$ costing 10, but A* would return the solution $start$–B–$goal$ costing 3. This shows that GC1 does not cost-dominate A*. In general, although globally compatible algorithms are required to return optimal solutions whenever A* does, they are allowed to return more costly solutions than A*'s when A*'s solutions are not optimal, as in this example.[5]

7 Results R4 and R5 (Dechter & Pearl's Corollary 7)

Dechter & Pearl's Corollary 7 also considers the situation in which the heuristics are admissible but not necessarily consistent but with algorithms restricted in a different way. This corollary focuses on best-first search algorithms, which are algorithms instantiating their BF* template using an evaluation function for paths, which we will denote $F(P)$. $F(P)$ computes a value for path P based on P's states, edge costs, and the heuristic values of P's states. We will write BF*(F) to denote BF* executed with the evaluation function F.

In the context of their Corollary 7 they require $F(P) = \omega(P)$ if P is a path from $start$ to $goal$ but otherwise impose no restrictions on F. A particular function F that satisfies this requirement is, of course, A*'s evaluation function: for path P from $start$ to n, $F(P) = \omega(P) \times h(n)$. We will use lower case f to refer to this evaluation function. A*, in this notation, is BF*(f).

Dechter & Pearl's Corollary 7, in our terminology, is this:[6]

> *For any heuristic h, non-pathological problem $\mathcal{P} \in \mathbb{P}_{AD}(h)$, and path evaluation function F such that BF*(F) succeeds on $\mathbb{P}_{AD}(h)$, if A^*_h breaks ties in favour of goal then any state expanded by A^*_h in solving $\mathcal{P}$ must also be expanded by BF*(F) in solving $\mathcal{P}$.*

Dechter & Pearl's Corollary 7 follows immediately from their Theorem 12(a), which is a mere restatement of their Lemma 6.[7] We shall focus, therefore, on their Lemma 6.

The following generalize $Q_f(P)$ (Definition 11) and Definition 8, respectively, to arbitrary path evaluation functions.

Definition 14. *For heuristic h, problem $\mathcal{P}$, path $P = \langle s_0, \ldots, s_n\rangle$, and path evaluation function F, define $Q_F(P) = F(s_0, \ldots, s_k)$, where $k \in [0, n]$ is such that $F(s_0, \ldots, s_k) \succeq F(s_0, \ldots, s_i)$ for all $i \in [0, n]$.*

Definition 15. *Let h be a heuristic, $\mathcal{P}$ a problem, and F a path evaluation function. Then path P is* strictly C^*-bounded *w.r.t. F iff $Q_F(P) \prec C^*$ and state s is* surely expanded *w.r.t. F (abbreviated F-s.e.) iff there is a path from start to s that is strictly C^*-bounded w.r.t. F.*

When $F = f$ this definition is identical to Definition 8 so states that we have been calling s.e. up to now will be called f-s.e. in the remainder of this section.

Dechter & Pearl's Lemma 6, in our notation, is as follows.

> *For any heuristic h and path evaluation function F, if BF*(F) succeeds on $\mathbb{P}_{AD}(h)$ then for every problem $\mathcal{P} \in \mathbb{P}_{AD}(h)$ any state that is f-s.e. is also F-s.e.*

[5] The flaw in Dechter & Pearl's proof of their Corollary 5 occurs in the proof that a *gc* algorithm must expand all s.e. states (their Theorem 11). That proof incorrectly asserts that A* will always find an optimal solution on their equivalent of $\mathcal{P}^s_{cd}$.

[6] Here we only give the portion of their Corollary 7 related to 0-optimality. The corollary also includes claims about 1-optimality.

[7] In their proof of their Theorem 12(a) they mistakenly cite their Theorem 11 when they meant to cite their Lemma 6.

Their proof is as follows (of course, they used the numerical operators, not the cost-algebraic ones).[8] Suppose, for the purpose of contradiction, that there exists a problem $\mathcal{P} \in \mathbb{P}_{AD}(h)$ in which some state s is f-s.e. but not F-s.e. Being f-s.e. means there is a path P from $start$ to s such that $Q_f(P) \prec C^*$, and not being F-s.e. means there is no path R from $start$ to s for which $Q_F(R) \prec C^*$. In particular, $Q_F(P) \succeq C^*$.

They then construct a problem $\mathcal{P}' \in \mathbb{P}_{AD}(h)$ that consists only of path P, $goal$,[9] and two edges to $goal$, one from s costing c_1, the other from $start$ costing c_2. Figure 2 depicts $\mathcal{P}'$. They then define c_1 and c_2 so that the following all hold:

(A) $(\omega(P) \times c_1) \prec c_2$,
(B) $Q_f(P) \preceq (\omega(P) \times c_1)$, and
(C) $c_2 \preceq Q_F(P)$.

With these facts, the logic of the proof is as follows. Because of (A), $C^*_{\mathcal{P}'} = \omega(P) \times c_1$, i.e. the only optimal solution in $\mathcal{P}'$ is P followed by the edge from s to $goal$. (B) guarantees that all the heuristic values are admissible, i.e. that $\mathcal{P}' \in \mathbb{P}_{AD}(h)$. Finally, (C) guarantees that the optimal path will not be found by BF*(F) because when it expands $start$ it finds a solution path costing c_2, so it will stop expanding states on P when it reaches the first one whose F-value equals $Q_F(P)$. Therefore, BF*(F) returns a suboptimal solution for $\mathcal{P}'$, contradicting the assumption that it succeeds on $\mathbb{P}_{AD}(h)$.

All that remains is to find values of c_1 and c_2 that satisfy (A), (B), and (C). Here Dechter & Pearl exploit the fact that their costs can be arbitrary positive numbers. They define

$$c_1 = Q_f(P) - c(P)$$

and

$$c_2 = \frac{Q_F(P) + Q_f(P)}{2}.$$

It is easy to verify that (A), (B), and (C) hold with these definitions. However, our cost algebra does not have operations equivalent to subtraction or division by 2, so these definitions cannot be used, something different is needed if this proof is to be used within the cost-algebraic setting.

Requirement (C) can always be satisfied by setting $c_2 = Q_F(P)$, which also gives the most latitude in satisfying (A).

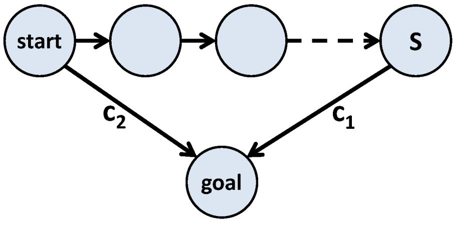

Figure 2: The construction in Dechter & Pearl's Lemma 6.

But requirements (A) and (B) are problematic in the cost-algebraic setting; a value of c_1 satisfying both requirements is not guaranteed to exist, as we will show in a moment.

If c_2 is assigned the value $Q_F(P)$ then setting $c_1 = h(s)$ always satisfies (A) since $\omega(P) \times h(s) = f(s) \preceq Q_f(P) \prec Q_F(P) = c_2$. If, in addition, $f(s) = Q_f(P)$, then (B) will be satisfied and the proof of the lemma completed. We therefore get Dechter & Pearl's Lemma 6, restricted to problems in $\mathbb{P}_{CONS}(h)$, in the cost-algebraic setting by making two final, simple observations:

(1) if the given problem $\mathcal{P}$ is in $\mathbb{P}_{CONS}(h)$ (as opposed to $\mathbb{P}_{AD}(h)$), then $\mathcal{P}'$ constructed as described (with $c_1 = h(s)$ and $c_2 = Q_F(P)$) is also in $\mathbb{P}_{CONS}(h)$, and

(2) in the cost-algebraic setting with a consistent heuristic f-values along a path can never decrease (just as in the traditional numerical setting).

Lemma 9. *For any heuristic h and path evaluation function F, if BF*(F) succeeds on $\mathbb{P}_{CONS}(h)$ then for every problem $\mathcal{P} \in \mathbb{P}_{CONS}(h)$ any state that is f-s.e. is also F-s.e.*

Combining Lemma 9 and Lemma 3, we get R5:

Theorem 10. *For any heuristic h, non-pathological problem $\mathcal{P} \in \mathbb{P}_{CONS}(h)$, and path evaluation function F such that BF*(F) succeeds on $\mathbb{P}_{CONS}(h)$, if A^*_h breaks ties in favour of goal then any state expanded by A^*_h in solving $\mathcal{P}$ must also be expanded by BF*(F) in solving $\mathcal{P}$.*

Counterexample to the cost-algebraic version of Dechter & Pearl's Lemma 6.

We will now give a counterexample showing that Dechter & Pearl's Lemma 6 is not true for $\mathbb{P}_{AD}(h)$ in the cost-algebraic setting. The algebra we will use in this example is defined as follows. The values of the algebra are all integers of the form $5a + 7b$ where a and b are any non-negative integers. For example, 22 is a value in this algebra ($22 = 5 \cdot 3 + 7 \cdot 1$) but 23 is not. The algebra's ordering $\preceq$ is the usual $\leq$ ordering on integers. The algebra's $\times$ operator is normal integer addition; 0 is the identity element ("**1**"). The set of values is closed under $\times$ because $(5a + 7b) + (5x + 7y) = 5(a + x) + 7(b + y)$.

The function $F(P)$ in this example computes the smallest cost possible for an extension of P that reaches $goal$ taking into account the g- and h-values of all the states in P. If the algebra contained all the non-negative integers this would simply be the largest f-value on P but because some integers are missing in this algebra, $F(P)$ can be strictly larger than the largest f-value on P and still be admissible. This is precisely what Dechter & Pearl's Lemma 6 says cannot happen (see footnote 8).

For example, consider the path $P = start$–A–B in Figure 3. $Q_f(P) = f(A) = 21$ but $F(P) = 22$ is admissible because the cost of P is 12 and it is impossible to extend a path of cost 12 with values in the algebra to create a path costing exactly 21. The cheapest extension of P to $goal$ that costs 21 or more has a cost of 22 so $F(P) = 22$ is admissible. In a problem where the only path from $start$ to $goal$ was path P followed by an edge from B to $goal$ costing 10, C^* would be 22, and B would be f-s.e. but not F-s.e. Yet BF*(F), for the F defined here, would succeed on $\mathbb{P}_{AD}(h)$.

[8] Their proof is not specific to s.e. states. What is actually being proven is that if BF*(F) succeeds on $\mathbb{P}_{AD}(h)$ then for every path P, $Q_F(P) \preceq Q_f(P)$.

[9] Strictly speaking, Dechter & Pearl add two new goal states, one for each edge. The difference is immaterial.

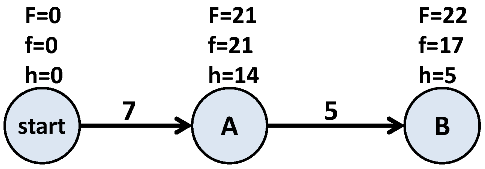

Figure 3: Counterexample to Dechter & Pearl's Lemma 6 in the cost-algebraic setting. $F(start) = 0, F(start - A) = 21$, and $F(start - A - B) = 22$.

8 Nilsson's RESULT 6

RESULT 6 (p. 81) in (Nilsson 1980)[10] is this: if heuristics h_1 and h_2 are admissible and $h_1(s) > h_2(s)$ on every non-goal state s then every state expanded by $A^*_{h_1}$ is also expanded by $A^*_{h_2}$. Unlike the Dechter & Pearl results we have examined in this paper, RESULT 6 is about the relative pruning power of heuristics and not about whether A* is optimally efficient with respect to other search algorithms. Nevertheless it is an intuitively appealing foundational result which, as we shall now show, does not hold in the cost-algebraic setting.

Given two heuristics, h_1 and h_2, we define $f^1_P(s) = g_P(s) \times h_1(s)$ and $f^2_P(s) = g_P(s) \times h_2(s)$ for any state s and any path P from $start$ to s. By $f^1(s)$ and $f^2(s)$, we refer to s's f-values on $Open$ at a specific point in time.

Lemma 11. *Let h_1 and h_2 be heuristics, $\mathcal{P}$ a problem, and s a non-goal state. If $h_1(s) \succ h_2(s)$, then $f^1_P(s) \succeq f^2_P(s)$ for every path P from start to s.*

It is crucial to note that isotonicity does not guarantee that $f^1_P(s) \succ f^2_P(s)$ when $h_1(s) \succ h_2(s)$, $f^1_P(s) = f^2_P(s)$ when $h_1(s) \succ h_2(s)$ is permitted unless the isotonicity is strict.

Definition 16. *A monoid $\langle \Omega, \times, \mathbf{1} \rangle$ with total order $\preceq$ on Ω is* strictly isotone *iff $\langle \Omega, \times, \mathbf{1} \rangle$ with $\preceq$ is isotone and, in addition, $a \prec b$ implies both $(a \times c) \prec (b \times c)$ and $(c \times a) \prec (c \times b)$ for all $a, b \in \Omega$ and all $c \in \Omega \setminus Zero$, where $Zero = \{\mathbf{0}\}$ if there exists an element $\mathbf{0} \in \Omega$ such that $a \times \mathbf{0} = \mathbf{0} \times a = \mathbf{0}$ for all $a \in \Omega$ (such an element is said to be "absorptive"), and $Zero = \emptyset$ otherwise.*

Edelkamp et al. (2005) did not insist on strict isotonicity because their theory did not require it and some of their applications, such as widest path, were not strictly isotone.

The cost-algebraic equivalent of Nilsson's RESULT 6 is easily proven when strict isotonicity holds.

Lemma 12. *Let $\langle \Omega, \times, \preceq, \mathbf{1} \rangle$ be a cost algebra, h_1 and h_2 heuristics, and $\mathcal{P} \in \mathbb{P}_{AD}(h_1) \cap \mathbb{P}_{AD}(h_2)$ a problem. If $\langle \Omega, \times, \mathbf{1} \rangle$ with $\preceq$ is strictly isotone and $h_1(s) \succ h_2(s)$ for every non-goal state s, then every state expanded by $A^*_{h_1}$ will also be expanded by $A^*_{h_2}$.*

When the isotonicity is not strict, RESULT 6 does not hold, as seen in Figure 4. Here the optimal path, costing C^*, is *start–B–goal*. Using the stronger heuristic, h_1, $f^1(A) = f^1(B)$ so $A^*_{h_1}$ must choose between them. Without loss of generality, let A be the one it chooses. A is a deadend, so $A^*_{h_1}$ will expand both A and B in solving this problem. With the weaker heuristic, h_2, $A^*_{h_2}$ will only expand B because $f^2(B) \prec f^2(A)$ and, once B is expanded, it will have a solution whose cost is $C^* = f^2(A)$. This example cannot happen with strict isotonicity because it would ensure that $f^1(A) \succ f^2(A)$, which would mean either $f^1(A) \succ C^*$, in which case $A^*_{h_1}$ would not expand A, or $f^2(A) \prec C^*$, in which case $A^*_{h_2}$ would expand both A and B.

9 Conclusion

One of the appealing properties of A* is its optimal efficiency in terms of the number of states expanded – under certain circumstances, one is guaranteed not to fare better when using any alternative to A*. While Dechter & Pearl (1985) provided such results for the natural but rather restrictive setting of traditional shortest path problems, we established formal guarantees of A*'s optimal efficiency for a much wider class of search problems.

Our results are of importance in several ways. Firstly, the general cost-algebraic setting studied above can model scenarios that may well arise in practice but that are not covered by the traditional framework, such as, e.g., the widest path problem. Secondly, we pointed out a mistake in a result in the classical literature, and provided an alternative result of a similar flavour. Thirdly, our formal results provide a much better understanding of known results on A*'s optimal efficiency in the traditional shortest path setting, since they reveal which specifics of the traditional setting are essential for maintaining A*'s optimal efficiency. For example, R4 turns out to be incorrect when dropping the specific requirement that all non-negative integers be allowed as values, and Nilsson's RESULT 6 relies on strict isotonicity. Knowing which requirements are of importance for which type of optimality result may potentially help guide the design of optimally efficient algorithms for new types of search problems.

10 Acknowledgements

Financial support for this research was in part provided by Canada's Natural Sciences and Engineering Research Council (NSERC) and by a grant to the first author from the Fac-

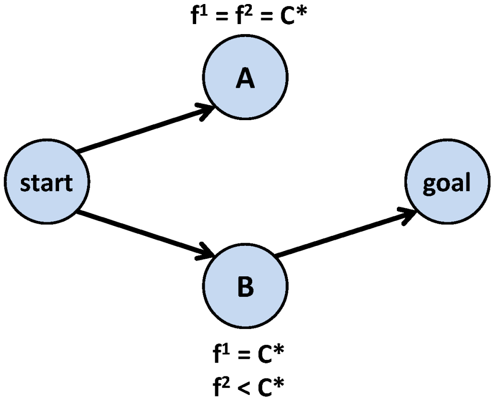

Figure 4: State A is expanded using the stronger heuristic (h_1) but not using the weaker heuristic (h_2).

[10] Theorem 3-2 (p. 64) in (Nilsson 1971) is essentially identical.

ulty of Science and the Computing Science Department of the University of Alberta.

Appendix A Results R1 and R2

Result R2 is a special case of R1, so we only need to show that R1 holds in the cost-algebraic setting. Our presentation of R1's proof is different than Dechter & Pearl's, but the proof itself is almost identical.

The basis for the proof of R1 is Dechter & Pearl's Theorem 8 (our Theorem 16). The key idea in proving this theorem is the construction of a new problem from a given problem $\mathcal{P}$ and s.e. state s. We call the new problem $\mathcal{P}^s$. It is identical to $\mathcal{P}$ except it has a new edge directly from s to $goal$ with a suitably chosen cost (c_s). Formally, $\mathcal{P}^s$ is defined as follows.

Definition 17. *Given heuristic* h*, problem* $\mathcal{P} = \langle start, goal, \sigma_{\mathcal{P}} \rangle \in \mathbb{P}_{CONS}(h)$*, and s.e. state* s*, define* $c_s = h(s)$*, and define a new problem* $\mathcal{P}^s = \langle start, goal, \sigma_{\mathcal{P}^s} \rangle$ *that is identical to* $\mathcal{P}$ *except for* $\sigma_{\mathcal{P}^s}(s)$*, which differs from* $\sigma_{\mathcal{P}}(s)$ *as follows:*

- *if* $goal \notin \sigma_{\mathcal{P}}(s)$ *then* $\sigma_{\mathcal{P}^s}(s) = \sigma_{\mathcal{P}}(s) \cup \{(goal, c_s)\}$,
- *if* $(goal, \omega_{\mathcal{P}}(s, goal)) \in \sigma_{\mathcal{P}}(s)$ *then* $\sigma_{\mathcal{P}^s}(s) = (\sigma_{\mathcal{P}}(s) \setminus \{(goal, \omega_{\mathcal{P}}(s, goal))\}) \cup \{(goal, c_s)\}$

Lemma 13. *Let* $h, \mathcal{P}, s, \mathcal{P}^s$*, and* c_s *be as in Definition 17. Then* $\mathcal{P}^s \in \mathbb{P}_{CONS}(h)$*.*

Lemma 14. *Let* $h, \mathcal{P}, s$*, and* c_s *be as in Definition 17. Then* $d(start, s) \times c_s \prec C^*_{\mathcal{P}}$*.*

Proof: Because s is s.e., there is a strictly $C^*_{\mathcal{P}}$-bounded path P from $start$ to s. Since s is a state on this path, we have $f_P(s) = \omega(P) \times h(s) \prec C^*_{\mathcal{P}}$. By the definition of $d(start, s)$, $d(start, s) \preceq \omega(P)$. Because the algebra is isotone, this implies $d(start, s) \times h(s) \preceq \omega(P) \times h(s)$. From $d(start, s) \times h(s) \preceq \omega(P) \times h(s)$ and $\omega(P) \times h(s) \prec C^*_{\mathcal{P}}$ it follows that $d(start, s) \times c_s \prec C^*_{\mathcal{P}}$. □

Corollary 15. *Let* $h, \mathcal{P}, s, \mathcal{P}^s$ *and* c_s *be as in Definition 17. Then* $C^*_{\mathcal{P}^s} \prec C^*_{\mathcal{P}}$*.*

Dechter & Pearl's Theorem 8 then follows.

Theorem 16. *For any heuristic* h*, problem* $\mathcal{P} = \langle start, goal, \sigma_{\mathcal{P}} \rangle \in \mathbb{P}_{CONS}(h)$*, and s.e. state* s*, any DXBB algorithm* A *that succeeds on* $\mathbb{P}_{CONS}(h)$ *must expand* s *in solving* $\mathcal{P}$*.*

Proof: Suppose, for the purpose of contradiction, that A succeeds on problem $\mathcal{P}$ without expanding s.e. state s and let X be the set of states expanded by A in solving $\mathcal{P}$. Since $s \notin X$ and s is the only state on which $\sigma_{\mathcal{P}^s}$ and $\sigma_{\mathcal{P}}$ differ, the premises of Lemma 6 are satisfied, so A will return the same solution for problem $\mathcal{P}^s$ as it did for $\mathcal{P}$. The cost of this solution is $C^*_{\mathcal{P}}$, but $C^*_{\mathcal{P}^s} \prec C^*_{\mathcal{P}}$ (Corollary 15), so A did not succeed on $\mathcal{P}^s$, contradicting the theorem's premise that A succeeds on $\mathbb{P}_{CONS}(h)$. □

Combining Theorem 16 and Lemma 3, we get R1.

Theorem 17. *For any heuristic* h*, non-pathological problem* $\mathcal{P} \in \mathbb{P}_{CONS}(h)$*, and DXBB algorithm* A *that succeeds on* $\mathbb{P}_{AD}(h)$*, if* A^*_h *breaks ties in favour of goal then any state expanded by* A^*_h *in solving* $\mathcal{P}$ *must also be expanded by* A *in solving* $\mathcal{P}$*.*

Algorithm 1: Generic DXBB Search Algorithm

```
Input: $\langle start, goal, \sigma \rangle, h$
Output: a least-cost path from $start$ to $goal$
$S_0 \leftarrow (\lambda, \{(start, 0)\})$
for $t$ from $1$ to $\infty$ do
    if StoppingCondition($t, S_0, \ldots, S_{t-1}, goal, h$) then
        return Solution($t, S_0, \ldots, S_{t-1}, goal$)
    $s_t \leftarrow$ Choose($t, S_0, \ldots, S_{t-1}, goal, h$)
    $S_t \leftarrow (s_t, \sigma(s_t))$
```

Appendix B DXBB Algorithms

Algorithm 1 is the template for DXBB algorithms by Eckerle et al. (2017) adapted to the present setting. A DXBB algorithm is any algorithm that can be implemented by instantiating `StoppingCondition`, `Solution`, and `Choose` in this template with deterministic functions.

On iteration t Algorithm 1 computes $S_t = (s_t, \sigma(s_t))$ recording the state s_t that was expanded and the results of that expansion (s_t must be chosen from among the states generated in previous iterations). S_0 is $(\lambda, \{(start, 0)\})$. $S_0, \ldots, S_{t-1}$ is called the expansion sequence up to iteration t. All the functions (`StoppingCondition`, `Solution`, and `Choose`) are given the expansion sequence but are not given σ, so they cannot peek ahead to see what lies beyond a state that has not yet been expanded. For more details see (Eckerle et al. 2017).

References

Bagchi, A., and Mahanti, A. 1983. Search algorithms under different kinds of heuristics – a comparative study. *J. ACM* 30(1):1–21.

Dechter, R., and Pearl, J. 1985. Generalized best-first search strategies and the optimality of A*. *J. ACM* 32(3):505–536.

Eckerle, J.; Chen, J.; Sturtevant, N. R.; Zilles, S.; and Holte, R. C. 2017. Sufficient conditions for node expansion in bidirectional heuristic search. In *Proc. 27th Intl. Conf. on Automated Planning and Scheduling (ICAPS)*, 79–87.

Edelkamp, S.; Jabbar, S.; and Lluch Lafuente, A. 2005. Cost-algebraic heuristic search. In *Proc. 20th National Conference on Artificial Intelligence (AAAI)*, 1362–1367.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE T. Syst. Sci. Cyb.* 4(2):100–107.

Nilsson, N. J. 1971. *Problem-Solving Methods in Artificial Intelligence*. McGraw-Hill.

Nilsson, N. J. 1980. *Principles of Artificial Intelligence*. Tioga Press.

Pearl, J. 1984. *Heuristics – Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.

Sobrinho, J. L. 2002. Algebra and algorithms for QoS path computation and hop-by-hop routing in the internet. *IEEE-ACM T. Network.* 10(4):541–550.