On the Optimal Efficiency of A* with Dominance Pruning

Submission #

Abstract

A well known result is that, given a consistent heuristic and no other source of information, A* does expand a minimal number of nodes up to tie-breaking. We extend this analysis for A* with dominance pruning, which exploits a dominance relation to eliminate some states during the search. We show that the expansion order of A* is not necessarily optimally efficient when considering dominance pruning with arbitrary dominance relations, but it remains optimally efficient under certain restrictions for the heuristic and dominance relation.

Introduction

Heuristic best-first search algorithms are a fundamental tool for problem solving whenever the problem can be modeled as finding paths in graphs. Heuristic functions guide the search towards the goal by estimating the distance from any given state to the goal. Whenever an optimal solution of minimum cost is required, A* search is often the algorithm of choice (Hart *et al.* 1968). This is well supported by the well known result that, given a consistent heuristic h and no other source of information, A* does expand a minimal number of nodes up to tie-breaking among all algorithms that guarantee finding the optimal solution (Dechter and Pearl 1985).

Dominance pruning is a technique to eliminate nodes during the search if they can be proven to be dominated by another state (Hall *et al.* 2013; Torralba and Hoffmann 2015). This exploits an additional source of information in the form of a dominance relation $\preceq$, which compares two states to determine whether one can be proven to be as close to the goal as the other. This type of dominance appears naturally on problems that have to deal with resources, (i.e., removing states that have strictly less resources than another), and can also be applied on other kinds of problems (e.g., in gridworlds being at a central square can sometimes be proven better than being at a corner if the set of reachable squares is strictly larger). This can be exploited by any search algorithm to reduce the number of nodes explored while retaining any solution optimally guarantees. This has been mainly used in the context of cost-optimal planning, as an enhancement for the A* algorithm, which is popular in that context.

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we address the question of whether the expansion order of A* is good to minimize the number of expansions when dominance pruning is used. Prioritizing the expansion of states with lower f-value is not necessarily an obvious choice anymore, since states that are more promising according to the heuristic function, are not necessarily better according to the dominance relation. Furthermore, previous results proving the optimal efficiency of A* are no longer valid due to having a new source of information.

Indeed, we show that there are cases where A* with dominance pruning is not optimally efficient, and that different expansion orderings, or even expanding some states that could be pruned lead to a globally number of expansions in some cases. However, these cases can be attributed to "inconsistencies" in the information provided by the heuristic function and the dominance relation. We extend the notion of consistent heuristics to consistent heuristic and dominance relation pairs, and prove that A* with dominance pruning is indeed optimally efficient (up to tie-breaking) when it is provided with consistent information.

Background

A *transition system* (TS) is a tuple $\Theta = \langle S, L, T, s^I, S^G \rangle$ where S is a finite set of *states*, L is a finite set of *labels* each associated with a *label cost* $c(l) \in \mathbb{R}_0^+$, $T \subseteq S \times L \times S$ is a set of *transitions*, $s^I \in S$ is the *start state*, and $S^G \subseteq S$ is the set of *goal states*. A *plan* for a state s is a path from s to any $s_G \in S^G$. We denote $h^*(s)$ ($g^*(s)$) to the cost of a cheapest plan for s (path from s^I to s). A plan for s is *optimal* iff its cost equals $h^*(s)$. We denote F^* to the cost of an optimal solution for s^I. To simplify the presentation, we assume all labels to have unit-cost. A *heuristic* $h : S \mapsto \mathbb{R}_0^+$ is a function that estimates goal distance. It is *admissible* if it never overestimates the real cost, i.e., $h(s) \leq h^*(s)$ for all $s \in S$, and *consistent* if for any $s \xrightarrow{l} t$ $h(s) - h(t) \leq c(l)$.

Best-first search algorithms maintain an open and a closed list with all nodes that have been seen so far. A search node n_s characterizes a path from the initial state as the final state of the path, s, where the g-value $g(n_s)$ is the cost of the path. The open list is initialized with the initial state that has a g-value of 0. At each step, a node is selected from the open list for expansion. When a node is expanded, it is removed from open and all the successors are generated and

inserted into the open list. The closed list keeps all nodes that have been expanded to avoid duplicates so that a node is not expanded if another node with the same state and a lower or equal g-value has already been expanded. A* always selects for expansion a node with minimum f-value where $f(n_s) = g(n_s) + h(s)$. Since the behavior of A* is not uniquely defined, we say that it is a family of algorithms, one per possible tie-breaking rule.

Optimal Efficiency of A* The seminal work by Dechter and Pearl (1985) analyze the optimal efficiency of A* in great depth, considering several degrees of optimal efficiency. They consider the heuristic as part of the input to the algorithm, so that a problem instance is a tuple $\langle \Theta, h \rangle$. An instance is consistent if it has a consistent heuristic h. The main result of interest for us is that, on consistent instances, there exists at least one tie-breaking for A* that expands a subset of the nodes expanded by any other admissible algorithm. It is important to mention that an important assumption that we also make is that the transition relation can only be accessed in a forward manner, as a function that given a state returns its successors. If backward search is possible, A* does not guarantee optimal efficiency (Chen *et al.* 2017).

Dominance Pruning Dominance pruning is a technique that makes use of a dominance relation as an additional source of information. A relation $\preceq \subseteq S \times S$ is a dominance relation if, whenever $s \preceq t$, then $h^*(t) \leq h^*(s)$.

We define A* with dominance pruning as the vanilla A* algorithm with a simple modification of A*. Anytime that a node n_s is selected for expansion, skip it if there exists another node n_t in open or closed such that $g(n_t) \leq g(n_s)$ and $s \preceq t$. Nodes pruned this way are removed from open but they are not expanded neither inserted into the closed list. Nodes can also be pruned upon generation to avoid the overhead of computing h and open list insertion. But this does not affect the number of expanded states, which interest us. We say that n_t prunes n_s if $g(n_t) \leq g(n_s)$ and $s \preceq t$.

Current methods for finding dominance relations (Torralba and Hoffmann 2015) obtain relations that are:

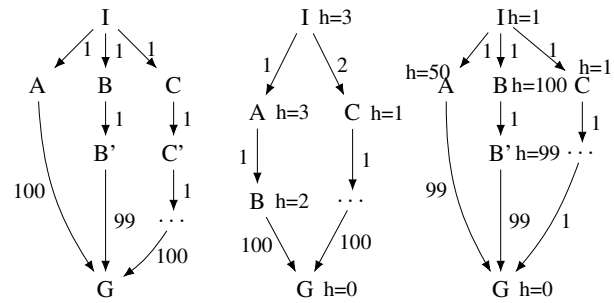
- Transitivity: $s \preceq t \wedge t \preceq u \implies s \preceq u$.
- Cost-simulation: whenever $s \preceq t$, for all $s \xrightarrow{l} s'$, either $s' \preceq t$ or $\exists t \xrightarrow{l'} t'$ s.t. $c(l') \leq c(l)$ and $s' \preceq t'$.

Counter-examples due to Inconsistencies

We begin considering the three reasons that may cause A* with dominance pruning to not be optimally efficient:

1. A node is pruned whenever possible, and sometimes not pruning a node may lead to less overall expansions.
2. The expansion order of A* may not be optimally efficient anymore when considering dominance pruning.
3. The information provided by the heuristic function and the dominance relation is not combined.

Figure 1 shows examples where A* with dominance pruning does expand more nodes than necessary for these three reasons. The example in Figure 1a illustrates a state space and dominance relation $\preceq$ for which pruning a node whenever possible is not an optimal strategy, independently of the expansion order (for simplicity we set $h = 0$ for all states). After expanding the initial state I, one can prune node B because it is dominated by A. However, if B is pruned, B' won't be generated under any expansion order so C' and all its arbitrarily many successors will be expanded.



(a) $\{(B, A), (C', B')\}$ (b) $\{(C, B)\}$ (c) $\{(C, B')\}$

Figure 1: Counterexamples that show cases where A* with dominance pruning is not optimally efficient, when pruning according to the dominance relation below each figure. The "..." region represents an arbitrarily large region of the state space that will be expanded by A* with dominance pruning, but could be avoided with a different pruning or expansion order strategies. In (a) $h = 0$ for all states, whereas in (b) and (c) each node is labeled with its heuristic value.

Our second example, illustrated in Figure 1b, shows a case where it is good to prune nodes whenever possible but the expansion order of A* leads to a sub-optimal number of expansions. The optimal expansion order is $\langle I, A, B, B', G \rangle$. C does not need to be expanded even though $f(C) < f^*$ because C will be pruned ($C \preceq B$ and $g(B) \leq g(C)$). However, A* with dominance pruning will expand C after expanding the initial state I, since $f(C) < f(A)$ and B has not been generated yet.

In the third example (see Figure 1c) shows an example where node C cannot be pruned due to B' because it has lower g-value. However, since h is admissible and $C \preceq B'$, then $h(B') \leq h^*(B') \leq h^*(C)$, so one can use $h(B')$ instead of $h(C)$, increasing the f-value of C and avoiding its expansion. However, A* will expand C before B, not noticing this. In general, one can use the information from the dominance relation to improve the heuristic values.

Definition 1. *The improved heuristic value is* $\bar{h}(s) = \max_{t \in S, s \preceq t} h(t)$. *The improved solution cost bound is then* $\bar{f}(n_s) = g(n_s) + \bar{h}(s)$.

Note that $\bar{h}$ dominates h because $\preceq$ is always reflexive.

All these three scenarios can be attributed to "inconsistencies" within the dominance relation $\preceq$ or between $\preceq$ and the heuristic function h. In the cases of Figure 1b and Figure 1c, the dominance relation and heuristic do not agree on the comparison between B/B' and C. The dominance relation proves that B is at least as close to the goal as C, but the heuristic function estimates that C is closer to the goal.

In the case of Figure 1a, the dominance relation is inconsistent because the information that A is closer to the goal than B is lost after one expansion and neither A nor any of its successors could be used to prune B' or C'.

Consistency of Heuristics and Dominance

A problem instance is a tuple $\langle \Theta, h, \preceq \rangle$, where Θ is a transition system, h is an admissible heuristic for Θ, and $\preceq$ is a dominance relation for Θ. We say that an instance is consistent if both the heuristic and dominance relation are consistent on their own, and they are consistent with each other.

Definition 2. *An instance $I = \langle \Theta, h, \preceq \rangle$ is consistent if:*

(i) *h is consistent.*

(ii) *$\preceq$ is a transitive cost-simulation.*

(iii) *$\preceq$ is consistent with h: $s \preceq t \implies h(t) \leq h(s)$.*

Condition (ii) ensures that the information provided by $\preceq$ is consistent in two different ways. First, $\preceq$ must be transitive, because if we do know that $s \preceq t$ and $t \preceq u$, then $h^*(u) \leq h^*(t) \leq h^*(s)$ so $s \preceq u$ holds as well. Second, for a dominance relation to be consistent, we require it to be a cost-simulation relation so that whenever $s \preceq t$, if s or any of its successors would prune u, then t or some of its successors prune u as well.

Condition (iii) requires $\preceq$ and h to not contradict each other on their comparison for any two states s and t. Note that this does not render $\preceq$ uninformative, since comparing states based on their heuristic value is no substitute for dominance analysis. In particular, even if $\preceq$ always agrees with h, its role is to identify cases where the relative heuristic evaluation of both states is provably correct.

A question is whether these conditions are extremely rare or they can be expected to happen in practice. The first two conditions are indeed quite common: most heuristics that come from an optimal solution to a relaxation of the problem are indeed consistent; and typical approaches to compute dominance relations in planning are guaranteed to return transitive cost-simulation relations (Torralba and Hoffmann 2015; Torralba 2017). An analysis of whether common planning heuristics are consistent with respect a dominance relation $\preceq$ is beyond the scope of this paper. However, in practice, it is reasonable to expect that many heuristics will fulfill this property, as has been shown to be the case for symmetry reduction techniques (Shleyfman *et al.* 2015).

Optimal Efficiency of A* with Dominance

Next, we prove that on consistent instances, A* with dominance pruning is guaranteed to be an optimally efficient algorithm. We first identify which states need to be expanded to prove optimality by any search that does not have access to any additional information, other than a heuristic function h and a dominance relation $\preceq$. Traditionally, this is done by identifying *must-expand* states that must be expanded for every algorithm to prove optimality, or *must-expand* pairs as done in the bidirectional search setting (Eckerle *et al.* 2017). However, in our case there are many choices that can be made for dominance pruning, as illustrated by our examples of Figure 1. We define instead solution sets, which take into consideration all nodes that must be expanded by A* to find a solution, including the last f-layer. Let S be a set of states. We denote $[S]$ to set extended with its immediate successors as , i.e., $[S] = S \cup \{s' \mid s \to s', s \in S\}$.

Definition 3. *Let I be an instance. A set of nodes $\mathcal{S}$ is a solution set for I if:*

(a) *$\forall_{n_s \in \mathcal{S}}$ either $s = s^I$ or $\exists_{n_t \in \mathcal{S}}$ such that $t \to s$, $g(n_s) = g(n_t) + 1$*

(b) *$\exists_{n_s \in [\mathcal{S}]} s \in S^G$ and $g(n_s) = f^*$,*

(c) *$\forall_{n_s \in [\mathcal{S}] \setminus \mathcal{S}}\ \bar{f}(n_s) \geq f^*$ or $\exists_{n_t \in \mathcal{S}} n_t$ prunes n_s.*

Condition (a) requires that every expanded node in S was generated by expanding one of its parents. Condition (b) requires that an optimal solution was found. Condition (c) ensures that the solution found is proven to be optimal, because all nodes in the open list after expanding $\mathcal{S}$ have a large enough f-value or are pruned by dominance.

Theorem 1. *Let I be any instance. Then, expanding a solution set is a necessary and sufficient condition to find a solution and prove optimality.*

Proof Sketch. If (a), (b), and (c) hold, then an optimal solution has been found due to (a) and (b), and the stopping condition holds, since all states remaining in the open list have an f-value greater or equal to the incumbent solution.

If (a) does not hold, then a state has been expanded without being in open, which is impossible in any best-first search algorithm. If (b) does not hold, then an optimal solution has not been found. If (c) does not hold, then there exists some s in the open list that may lead to a solution cost lower than f^*, so the solution has not proven to be optimal. □

We start by proving some properties that always hold on consistent instances. We begin by showing that, on consistent instances, the dominance relation cannot be used to improve the heuristic. Based on this, we will ignore the difference between $\bar{f}$ and f on the rest of the section.

Lemma 1. *Let I be an instance where h is consistent with $\preceq$. Then, $h = \bar{h}$, so $f = \bar{f}$.*

Proof. $\bar{h} = \max_{t \in S, s \preceq t} h(t)$. As $s \preceq t$ implies $h(t) \leq h(s)$, the maximum of this expression is always $h(s)$. □

An important result is that, whenever h and $\preceq$ are consistent with each other, nodes with larger f-value cannot prune nodes with lower f-value.

Lemma 2. *Let I be a consistent instance. Let n_s, n_t be any two nodes such that n_t prunes n_s. Then, $f(n_t) \leq f(n_s)$.*

Proof. Since n_t prunes n_s, it holds that $g(n_t) \leq g(n_s)$ and $s \preceq t$. By consistency, $h(t) \leq h(s)$, so $f(n_t) \leq f(n_s)$. □

Next, we show that pruning is transitive.

Lemma 3. *If $\preceq$ is transitive, n_u prunes n_t and n_t prunes n_s, then n_u prunes n_s.*

Proof. By the assumption it follows that $g(n_u) \leq g(n_t) \leq g(n_s)$, and $s \preceq t \preceq u$. Therefore, $g(n_u) \leq g(n_s)$ and, by transitivity of $\preceq$, $s \preceq u$. So n_u prunes n_s. □

Next, we show that all states in the smallest solution set must be expanded only with its optimal g-value.

Lemma 4. *Let I be a consistent instance. Then, there exists a solution set $\mathcal{S}$ for I of minimum size such that for all $n_s \in \mathcal{S}$, $g(n_s) = g^*(s)$.*

Proof. Assume the contrary. Then, some n_s has been expanded with a sub-optimal value, $g^*(s) < g(n_s)$. Therefore, a predecessor along the optimal path from s^I to s has not been expanded. Let n_t be the first such predecessor. By consistency of h, we know that f-values monotonically increase along a path, so $f(n_t) \leq f^*(n_s) < f(n_s)$. As $n_t \notin \mathcal{S}$. By condition (c) of a solution set, n_t was pruned, i.e., $\exists n_u \in \mathcal{S}$ s.t. n_u prunes n_t. As $\preceq$ is a cost-simulation, then n_u has some successor that would prune n_s, so there must be a node in $\mathcal{S}$ that prunes n_s. Therefore, $\mathcal{S} \setminus \{n_s\}$ is also a solution set, contradicting the fact that $\mathcal{S}$ is of minimum size. □

Lemma 5. *Let $\mathcal{S}$ be a solution set for a consistent instance. Let $n_s \in \mathcal{S}, n_t \in [\mathcal{S}]$ such that n_t prunes n_s. Then, there exists another solution set $\mathcal{S}'$ that does not contain n_s such that $|\mathcal{S}'| < |\mathcal{S} \cup \{n_t\}|$.*

Proof. We construct $\mathcal{S}'$ as follows. Initialize $\mathcal{S}'$ as the maximal subset of $(\mathcal{S} \setminus \{n_s\}) \cup \{n_t\}$ such that condition (a) holds (i.e. insert n_t and then remove n_s from $\mathcal{S}$ along with any other node in $\mathcal{S}$ that was only reachable through s). As $\preceq$ is a cost-simulation, for each node $n_{s'}$ removed in this way, there exists $n_{t'}$ reachable from n_t s.t. $s' \preceq t'$, so add all those $n_{t'}$ to $\mathcal{S}'$. Note that at most one such $n_{t'}$ node is inserted per each $n_{s'}$ removed so $|\mathcal{S}_{i+1}| \leq |\mathcal{S}_i|$. We know that $\mathcal{S}'$ is a solution set: condition (c) holds because for any n_u that was pruned due to some $n_{s'}$, by transitivity (see Lemma 3), it is pruned now by some of the newly added $n_{t'}$. □

We use the previous Lemma to show that: (1) pruning a node whenever possible is an optimally efficient strategy because there is a solution set of minimum size such that no state that could be pruned was expanded; and (2) if n_t prunes n_s it can never be wrong to expand n_t instead of n_s.

Lemma 6. *Let $\mathcal{S}$ be a solution set of minimum size for a consistent instance. Then there do not exist n_s, n_t in $\mathcal{S}$ such that n_t prunes n_s.*

Proof. Assume that n_t prunes n_s. By Lemma 5, there is another solution set $\mathcal{S}'$ such that $|\mathcal{S}'| < |\mathcal{S} \cup \{n_t\}|$. As $n_t \in \mathcal{S}$, $|\mathcal{S}'| < |\mathcal{S}|$, contradicting that $\mathcal{S}$ has minimum size. □

Lemma 7. *Let I be a consistent instance. Then, there exists a minimal solution set $\mathcal{S}$ for I such that there does not exist any $n_s \in \mathcal{S}$ and $n_t \in [\mathcal{S}] \setminus \mathcal{S}$ such that n_s prunes $n_t$and n_t does not prune n_s.*

Proof. We construct $\mathcal{S}$ as follows. Let $\mathcal{S}_0$ be any minimal solution set. Then repeat the following process. Let $n_s \in \mathcal{S}_i$ and $n_t \notin \mathcal{S}_i$ such that n_t prunes n_s and $t \not\preceq s$. Choose such an n_s with minimal g-value. Apply Lemma 5 to obtain a new $\mathcal{S}_{i+1}$ without n_s, replacing it by n_t. As $\preceq$ is a partial order, repeating this process iteratively will converge replacing each n_s by some n_t until there does not exist any $n_u \in [\mathcal{S}_i] \setminus \mathcal{S}_i$ such that $t \preceq u$ and $u \not\preceq t$. □

Now we are ready to prove our main result.

Theorem 2. *Let I be a consistent instance. Then A* with dominance pruning is optimally efficient up to tie-breaking.*

Proof. We show that there exists a minimal solution state set $\mathcal{S}$ for which there exists a tie-break under which A* with h and $\preceq$ expands exactly $\mathcal{S}$. By Lemma , we choose a solution set so that there does not exist any $s \in \mathcal{S}$ and $t \in [\mathcal{S}] \setminus \mathcal{S}$ such that $s \preceq t$ and $t \not\preceq s$. Assume a tie-break policy that prefers expanding states in $\mathcal{S}$ over any other state, and prefers pruning states not in $\mathcal{S}$. Formally, our tie-breaking strategy selects for expansion any node not in $\mathcal{S}$ such that it can be pruned. If no such state exists, it selects a state (with minimal f value) from $\mathcal{S}$ such that it cannot be pruned.

We show that this tie-breaking always succeeds. Otherwise, the open list does not contain any state with minimal f value that is outside $\mathcal{S}$ and can be pruned or that it is in $\mathcal{S}$ and cannot be pruned. Then, the state selected for expansion either: (A) it is in $\mathcal{S}$ but can be pruned due to some node in open or closed; (B) it is not in $\mathcal{S}$ and cannot be pruned.

Case (A). There exists n_t that prunes some $n_s \in \mathcal{S}$. By Lemma 6 we know that $n_t \notin \mathcal{S}$. As n_t is in the open list, $n_t \in [\mathcal{S}]$ and, by Lemma , it follows that $n_t \preceq n_s$. By Lemma 2, $f(n_t) = f(n_s)$, so our tie-breaking strategy would have selected n_t instead, reaching a contradiction.

Case (B). Let s be a state that is expanded by A* with dominance pruning but it is not in $\mathcal{S}$. If $f(n_s) = f^*$, then a node along the optimal solution contained in $\mathcal{S}$ should have been chosen instead. If $f(n_s) < f^*$, by condition (c) of a solution set, there exists $n_t \in \mathcal{S}$ such that n_t prunes n_s. Again, if n_t is in open or closed, n_s will be pruned reaching a contradiction. Otherwise, there must be a predecessor of n_t in open with its optimal g-value. Such an $n_u \in \mathcal{S}$, must have $f(n_u) \leq f(n_t) \leq f(n_s)$, so n_u would have been chosen for expansion instead of n_s. □

Conclusions

We analyzed the optimal efficiency of A* with dominance pruning. Assuming a consistent heuristic is not sufficient, because there may be inconsistencies in the dominance relation as well. We define a new criteria of consistency for heuristic and dominance relation pairs, which ensures that A* will be optimally efficient in terms of expanded nodes.

As in the optimal efficiency result for A*, our analysis is based only on the number of state expansions and it ignores the actual runtime. In fact, other algorithms may have superior runtime if the benefits of reducing the state expansion does not compensate the overhead of computing the heuristic. In our case, this could be more severe since checking dominance pruning may require a quadratic cost in the number of states in the worst case. Nevertheless, for concrete problems and/or dominance relations it may be possible to perform the pruning more efficiently (e.g., dividing states in classes so that each state needs to be compared only against a small subset of alternatives), and one could extend rational algorithms that reason about when it is worth to compute the heuristic (Karpas *et al.* 2018) to consider dominance as well.

References

Jingwei Chen, Robert C. Holte, Sandra Zilles, and Nathan R. Sturtevant. Front-to-end bidirectional heuristic search with near-optimal node expansions. In Carles Sierra, editor, *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pages 489–495. AAAI Press/IJCAI, 2017.

Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of A*. *Journal of the Association for Computing Machinery*, 32(3):505–536, 1985.

Jürgen Eckerle, Jingwei Chen, Nathan R. Sturtevant, Sandra Zilles, and Robert C. Holte. Sufficient conditions for node expansion in bidirectional heuristic search. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS'17)*, pages 79–87. AAAI Press, 2017.

David Hall, Alon Cohen, David Burkett, and Dan Klein. Faster optimal planning with partial-order pruning. In Daniel Borrajo, Simone Fratini, Subbarao Kambhampati, and Angelo Oddi, editors, *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*, Rome, Italy, 2013. AAAI Press.

Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Erez Karpas, Oded Betzalel, Solomon Eyal Shimony, David Tolpin, and Ariel Felner. Rational deployment of multiple heuristics in optimal state-space search. *Artif. Intell.*, 256:181–210, 2018.

Alexander Shleyfman, Michael Katz, Malte Helmert, Silvan Sievers, and Martin Wehrle. Heuristics and symmetries in classical planning. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, pages 3371–3377. AAAI Press, January 2015.

Álvaro Torralba and Jörg Hoffmann. Simulation-based admissible dominance pruning. In Qiang Yang, editor, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*, pages 1689–1695. AAAI Press/IJCAI, 2015.

Álvaro Torralba. From qualitative to quantitative dominance pruning for optimal planning. In Carles Sierra, editor, *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, pages 4426–4432. AAAI Press/IJCAI, 2017.