From Qualitative to Quantitative Dominance Pruning for Optimal Planning

Álvaro Torralba
Saarland University
Saarland Informatics Campus
Saarbrücken, Germany
torralba@cs.uni-saarland.de

Abstract

Dominance relations compare states to determine whether one is at least as good as another in terms of their goal distance. We generalize these qualitative yes/no relations to functions that measure by how much a state is better than another. This allows us to distinguish cases where the state is strictly closer to the goal. Moreover, we may obtain a bound on the difference in goal distance between two states even if there is no qualitative dominance.

We analyze the multiple advantages that quantitative dominance has, like discovering coarser dominance relations, or trading dominance by g-value. Moreover, quantitative dominance can also be used to prove that an action starts an optimal plan from a given state. We introduce a novel action selection pruning that uses this to prune any other successor. Results show that quantitative dominance pruning greatly reduces the search space, significantly increasing the planners' performance.

Introduction

Most classical planners focus on reducing the search space. Their success greatly depends on their ability to exploit the structure of the problem in the form of heuristics or pruning methods. Pruning methods reduce the search effort by eliminating redundant states (Pochter, Zohar, and Rosenschein 2011) or avoiding the application of some actions (Wehrle and Helmert 2012) while preserving at least one optimal plan. Dominance pruning methods automatically construct a relation that compares states, to eliminate those that are dominated by others. Previous approaches define a qualitative relation, $\preceq$, in which t is said to dominate s ($s \preceq t$) if it is at least as close to the goal (Hall et al. 2013). In that case, s may be safely pruned if its g-value is not lower than that of t.

We generalize the label-dominance (LD) simulation method originally devised to compute qualitative dominance (Torralba and Hoffmann 2015) to a quantitative version. Instead of a relation, we define a function $\mathcal{D} : S \times S \to \mathbb{R} \cup \{-\infty\}$ that measures "by how much" does t dominate s. A positive value $\mathcal{D}(s,t) > 0$ means that t is strictly closer to the goal than s. Negative values bound the difference in goal distance between t and s.

Theoretically, quantitative dominance has several advantages. First, it may find coarser relations, hereby strengthening previous dominance pruning methods. Second, and more importantly, novel pruning methods may take advantage of the additional information. One way is to trade-off dominance and g-value. If $\mathcal{D}(s,t) > 0$ we may prune s even if its g-value is lower. If $\mathcal{D}(s,t) < 0$ there is no qualitative dominance but, we can still prune s if its g-value is large enough. Another way is to use quantitative dominance to prove that an action a starts an optimal plan from a given state s, whenever the successor dominates s by an amount equal to the action cost. We introduce a novel type of pruning, which we call action selection pruning, that prunes any other successor reducing the branching factor to one.

Empirically, we show that quantitative dominance can greatly reduce the search space in many benchmark domains, even when compared to the qualitative version. However, there is a big overhead to perform as much pruning as possible so approximation methods may be desirable. Action selection, on the other hand, achieves an impressive amount of pruning with very low overhead. Moreover, it is complementary to previous dominance pruning methods and it greatly improves their performance in many domains.

Background

A *planning task* is a tuple $\Pi = \langle \mathcal{V}, \mathcal{A}, \mathcal{I}, \mathcal{G} \rangle$. $\mathcal{V}$ is a finite set of *variables* v, each with a finite domain D_v. A *partial state* is a function s on a subset $\mathcal{V}(s)$ of $\mathcal{V}$, so that $s(v) \in D_v$ for all $v \in \mathcal{V}(s)$; s is a *state* if $\mathcal{V}(s) = \mathcal{V}$. $\mathcal{I}$ is the *initial state* and the *goal* $\mathcal{G}$ is a partial state. $\mathcal{A}$ is a finite set of *actions*. Each $a \in \mathcal{A}$ is a tuple $\langle pre_a, \mathit{eff}_a, c_a \rangle$ where pre_a and eff_a are partial states, called its *precondition* and *effect*, and $c(a) \in \mathbb{R}^+_0$ is its cost. An action a is applicable in a state s if $s(v) = pre_a(v)\ \forall v \in \mathcal{V}(pre_a)$. In that case, the result of applying a in s, denoted $s[\![a]\!]$, is another state s.t. $s[\![a]\!](v) = \mathit{eff}_a(v)$ if $v \in \mathcal{V}(\mathit{eff}_a)$, and $s[\![a]\!](v) = s(v)$ otherwise.

A *labeled transition system* (LTS) 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 planning task defines a *state space*, which is an LTS where: S is the set of all states; $s^I = \mathcal{I}$; $s \in S^G$ iff $\mathcal{G} \subseteq s$; $L = \mathcal{A}$, and $s \xrightarrow{a} s[\![a]\!] \in T$ if a is applicable in s. We will use $s \in \Theta$ to refer to states in Θ and $s \xrightarrow{a} t$ to refer to their transitions.

A *plan* for a state s is a path from s to any $s_G \in S^G$. The cost of a cheapest plan for s is denoted $h^*(s)$, and the cost of a cheapest path from $\mathcal{I}$ to s is denoted $g^*(s)$. A plan for s is *optimal* iff its cost equals $h^*(s)$ and is *strongly optimal* if its number of 0-cost actions (denoted $h^{*0}(s)$) is minimal among all optimal plans for s.

We consider a representation of the planning task as a set of LTSs on a common set of labels, $\{\Theta_1, \dots, \Theta_k\}$ (Helmert, Haslum, and Hoffmann 2007; Helmert et al. 2014). Whenever it is not clear from the context, we will use subscripts to differentiate states in the state space, Θ (s, s', t) and in the individual components Θ_i (s_i, s_i', t_i). The synchronized product of two LTSs $\Theta_1 \otimes \Theta_2$ is another LTS with states $S = \{(s_1, s_2) \mid s_1 \in \Theta_1 \wedge s_2 \in \Theta_2\}$, transitions $T = \{(s_1, s_2) \xrightarrow{l} (s_1' s_2') \mid s_1 \xrightarrow{l} s_1' \wedge s_2 \xrightarrow{l} s_2'\}$, s.t. $(s_1, s_2) \in S^G$ iff $s_1 \in S_1^G$ and $s_2 \in S_2^G$.

Simulation-Based Qualitative Dominance

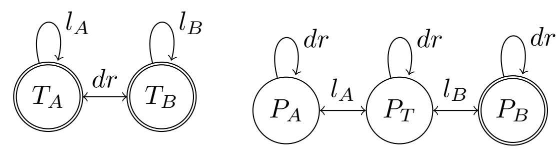
This section describes the label-dominance (LD) simulation method we build upon (Torralba and Hoffmann 2015). Given a planning task with states S, a *dominance relation* is a relation $\preceq \subseteq S \times S$ where $s \preceq t$ implies $h^*(t) < h^*(s)$ or $h^*(t) = h^*(s)$ and $h^{*0}(t) \leq h^{*0}(s)$. Such relation can be used to prune states during the search: A search node n_s (representing state s) can be pruned at any point if there exists a node $n_t \in$ *open* $\cup$ *closed* s.t. $g(n_t) \leq g(n_s)$ and $s \preceq t$.

A relation $\preceq$ is *goal-respecting* if whenever $s \preceq t$, $t \in S^G \vee s \notin S^G$. $\preceq$ is a *simulation* relation if, whenever $s \preceq t$, for every $s \xrightarrow{l} s'$, there exists a transition $t \xrightarrow{l} t'$ s.t. $s' \preceq t'$. A *cost-simulation* allows the transition from t to use a different label of lower or equal cost, i.e., whenever $s \preceq t$, for every $s \xrightarrow{l} s'$, there exists a transition $t \xrightarrow{l'} t'$ s.t. $s' \preceq t'$ and $c(l') \leq c(l)$.

In a compositional approach, we take as input a set of LTSs $\{\Theta_1, \dots, \Theta_k\}$ and compute a relation $\preceq_i$ on each Θ_i to obtain a goal-respecting cost-simulation of the whole state space $\Theta_1 \otimes \dots \otimes \Theta_k$. LD simulation computes all of them simultaneously, using label dominance to ensure that the property still holds after merging every Θ_i.

Definition 1 (LD Simulation) *A set $\{\preceq_1, \dots, \preceq_k\}$ of relations $\preceq_i \subseteq S_i \times S_i$ is a* label-dominance (LD) simulation *for $\{\Theta_1, \dots, \Theta_k\}$ if all $\preceq_i$ are goal-respecting and, whenever $s \preceq_i t$, for all $s \xrightarrow{l} s' \in \Theta_i$, there exists a transition $t \xrightarrow{l'} t'$ in Θ_i s.t. $s' \preceq_i t'$, $c(l') \leq c(l)$, and for all $j \neq i$, l' dominates l in Θ_j given $\preceq_j$. We say that l' dominates l in Θ_j given $\preceq_j$ if for all $s \xrightarrow{l} s' \in \Theta_j$ there exists $s \xrightarrow{l'} t' \in \Theta_j$ s.t. $s' \preceq t'$.*

Intuitively, t dominates s in Θ_i if, for every outgoing transition from s, t has an at least as good transition where the targets are compared according to $\preceq_i$ and the labels are compared in all other Θ_j to ensure that there is no negative side effect. For any LD simulation $\{\preceq_1, \dots, \preceq_k\}$, we can define a relation $\preceq$ s.t. $s \preceq t$ iff $s_i \preceq_i t_i$ for each Θ_i. This relation is a goal-respecting cost-simulation and hence, a valid dominance relation for the state space $\Theta \equiv \Theta_1 \otimes \dots \otimes \Theta_k$.



(a) Θ_1 (truck)

(b) Θ_2 (package)

Figure 1: LTSs describing our logistics running example.

A typical example is a logistics task where a single truck must transport n packages from location A to B. Figure 1 shows the LTSs of the case with a single package. In this example, LD simulation finds a relation where $P_A \preceq P_T \preceq P_B$, i.e., having a package at its destination is at least as good as having it in the truck, which is at least as good as having it anywhere else. This holds independently of the position of the truck or the other packages in case there are any. This allows to prune, for example, state $\langle T_A, P_A \rangle$ if $\langle T_A, P_T \rangle$ has lower or equal g-value. This is quite useful, as it prunes away any state in which a package has been unloaded in any location other than its destination. However, in the next sections we see that quantitative dominance can do much more.

Quantitative Dominance

First, we generalize the definition of dominance relations.

Definition 2 (Quantitative Dominance Function) *A function $\mathcal{D} : S \times S \to \mathbb{R} \cup \{-\infty\}$ is a quantitative dominance function for an LTS Θ iff $\mathcal{D}(s,t) \leq h^*(s) - h^*(t)$ and, if $h^*(s) = h^*(t)$ and $h^{*0}(s) < h^{*0}(t)$, then $\mathcal{D}(s,t) < 0$.*

Intuitively, if $\mathcal{D}(s,t) > 0$, then t is strictly closer to the goal than s; if $\mathcal{D}(s,t) = 0$ then t is at least as close to the goal as s; and if $-\infty < \mathcal{D}(s,t) < 0$, t can get as close to the goal as s by paying a price of $-\mathcal{D}(s,t)$. Finally, if $\mathcal{D}(s,t) = -\infty$, we did not discover any dominance of t over s. The second part of the definition ensures that the pruning is safe in domains with 0-cost actions, where s should not be dominated by t if it is in the path from t to the goal. Given a function $\mathcal{D}$, we can define dominance relations based on it.

Definition 3 (Quantitative Dominance Relation) *Let $\mathcal{D}$ be a quantitative dominance function on an LTS Θ and let $C \in \mathbb{R}$ be a constant. We define the C-dominance relation as $s \preceq_{\mathcal{D}}^{C} t$ iff $\mathcal{D}(s,t) \geq C$.*

This generalizes qualitative dominance, since $\preceq_{\mathcal{D}}^{0}$ is a qualitative dominance relation. For any other $\preceq_{\mathcal{D}}^{C}$, we distinguish between *positive* and *negative* dominance relations depending on whether $C > 0$ or $C < 0$. For unspecified C, $s \preceq_{\mathcal{D}}^{C} t$ serves as a shorthand for $\mathcal{D}(s,t) > -\infty$.

Quantitative Compositional LD Simulation

We follow a compositional approach. Given a set of LTSs $\{\Theta_1, \dots, \Theta_k\}$, we define a quantitative dominance for each of them so that their aggregation is a quantitative dominance function of the state space of the planning task, $\Theta_1 \otimes \ldots \otimes \Theta_k$.

To operationalize this definition, we draw upon LD simulation relations. Let s and t be two states for which $s \preceq t$. Then, in the standard notion of simulation any plan π_s for s must also be a plan for t. As this is too restrictive for deriving useful dominance relations, LD simulation allows to use different labels in the plan π_t from t and, if a noop action is considered, π_t can be shorter than π_s. A limitation is that it still requires the plan for t not to be longer than that from s. This is fine in qualitative dominance because there is usually a strong correlation between plan cost and length (Radzi 2011). However, it is an impediment to infer negative dominance since if there exists a path $t \rightarrow^* s$ of cost c we would like to infer that $\mathcal{D}(s,t) \geq -c$. Consider the position of the truck in our example. In an LD simulation, $T_A \not\preceq_1 T_B$ because of the transition $T_A \xrightarrow{l_A} T_A$ for which T_B does not have any counterpart (*noop* or l_B do not dominate l_A in the other LTSs). However, since the movements of the truck do not depend on any other variable, $\mathcal{D}_1(T_A, T_B) = -1$ because from T_B we can always reach T_A without having any side effects on other variables.

We avoid this restriction by considering weak simulation relations (Hennessy and Milner 1985). Weak simulations consider a set of internal τ-labels that are not relevant to describe the behavior of the system. Therefore, each transition $s \xrightarrow{l} s'$ can be simulated by a path $t \xrightarrow{\tau}{}^* u \xrightarrow{l} u' \xrightarrow{\tau}{}^* t'$ s.t. $s' \preceq t'$. In our case, τ-labels are those that do not have any preconditions or effects in other LTSs, like dr for the position of the truck in our example.

Definition 4 (τ-label) *Let $\{\Theta_1, \ldots, \Theta_k\}$ be a set of LTSs. Label l is a τ-label for Θ_i iff $s \xrightarrow{l} s \in \Theta_j$ $\forall \Theta_j \neq \Theta_i, s \in \Theta_j$.*

The particular actions in a τ-path are not relevant, only its cost is. We model this by defining the τ-distance between any two states.

Definition 5 (τ-distance) *Let s and t be two states in an LTS Θ. The, τ-distance from s to t, written $h^\tau(s,t)$, is the cost of a minimum-cost path from s to t in Θ using only transitions with τ labels or ∞ if no such path exists. 0-cost transitions are considered to have an infinitesimal cost ϵ.*

A non-goal state can only dominate a goal state if it has a τ-path to the goal, so we define a goal-respecting function in terms of the τ-distance.

Definition 6 (Goal-respecting function) *A function $\mathcal{D}$ is goal-respecting for Θ iff for all $s \in S^G$ and $t \in S$, $\mathcal{D}(s,t) \leq \max_{s_g \in S^G} -h^\tau(t, s_g)$.*

Finally, we extend the definition of label dominance to the quantitative case, by defining a function $\mathcal{D}^L(l,l')$ that captures the relation between labels.

Definition 7 (Label-dominance function) *Let $\mathcal{D}$ be a function for Θ, we define its corresponding label-dominance function as $\mathcal{D}^L(l,l') = \min_{s \xrightarrow{l} s' \in \Theta} \max_{s \xrightarrow{l'} s'' \in \Theta} \mathcal{D}(s', s'')$*

If $\mathcal{D}^L(l,l') > 0$, then every time that we can apply l in any state s, applying l' will lead us to a better state. If $-\infty < \mathcal{D}^L_j(l,l') < 0$, we could reach an at least as good state if we pay the corresponding price.

Definition 8 (QLD Simulation) *Let $\mathcal{D}_\mathcal{F} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ be a set of goal-respecting functions for $\mathcal{T} = \{\Theta_1, \ldots, \Theta_k\}$. $\mathcal{D}_\mathcal{F}$ is a* quantitative label-dominance (QLD) simulation *for $\mathcal{T}$ if for all $\Theta_i \in \mathcal{T}$ and $s, t \in \Theta_i$, $\mathcal{D}_i(s,t) \leq f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, s, t)$ where $f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, s, t) :=$*

$$\min_{s \xrightarrow{l} s'} \max_{u \xrightarrow{l'} u'} \mathcal{D}_i(s', u') - h^\tau(t,u) + c(l) - c(l') + \sum_{j \neq i} \mathcal{D}^L_j(l,l')$$

where $s \xrightarrow{l} s' \in \Theta_i, u \xrightarrow{l'} u' \in \Theta_i$

Intuitively, we compare all transitions from s ($s \xrightarrow{l} s'$), against the best alternative from t ($t \xrightarrow{\tau}{}^* u \xrightarrow{l} u'$)[1] by summing up the difference in goal-distance of the targets ($\mathcal{D}_i(s', u')$), the cost of the transition from s ($c(l)$), minus the cost that it takes to apply the transition from t ($h^\tau(t,u) + c(l')$). Finally, $\sum_{j \neq i} \mathcal{D}^L_j(l,l')$ estimates the benefit or penalty for using l' instead of l in the other LTSs. Applying this definition to our example, we now find some dominance for the truck $\mathcal{D}_1(T_A, T_B) = \mathcal{D}_1(T_B, T_A) = -1$. For the package, we find that $\mathcal{D}_2(P_A, P_T) = 1$, $\mathcal{D}_2(P_T, P_B) = 1$ so $\mathcal{D}_2(P_A, P_B) = 2$. This is similar to the result of LD simulation $P_A \preceq P_T \preceq P_B$, but with the additional information that is strictly closer instead of at least as close to the goal.

Theorem 1 *A unique maximal QLD simulation always exists.*

Proof Sketch: An QLD simulation always exists because the "identity" function s.t. $\mathcal{D}_i(s_i, t_i) = -\infty$ if $s_i \neq t_i$ and 0 otherwise is always an QLD simulation. Given any two QLD simulations, their maximum is also an QLD simulation so a unique maximal simulation exists. $\square$

Theorem 2 *Let $\mathcal{D}_\mathcal{F} = \{\mathcal{D}_1, \ldots, \mathcal{D}_k\}$ be an QLD simulation on $\mathcal{T} = \{\Theta_1, \ldots, \Theta_k\}$. Then, $\mathcal{D}_1 + \cdots + \mathcal{D}_k$ is a quantitative dominance function on $\Theta_1 \otimes \ldots \otimes \Theta_k$.*

A proof is included in the appendix.

Computing Quantitative LD Simulations

Algorithm 1 shows how to compute an QLD simulation for a set of LTSs $\mathcal{T}$, given a parameter, $\mathcal{K}$. Each $\mathcal{D}_i$ is initialized as the maximal goal-respecting function. Then, at each iteration it checks whether the property $\mathcal{D}_i(s,t) \leq f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, s, t)$ is violated for some $\mathcal{D}_i(s,t)$. In that case, it updates the value and repeats until the result is a valid QLD simulation. For sufficiently large $\mathcal{K}$ (e.g., if $\mathcal{K}$ is greater than the maximum cost of any plan of the task, which can be easily bounded by $|\Theta_1 \otimes \ldots \otimes \Theta_k|(\max_{l \in L} c(l))$), Algorithm 1 will find the maximal QLD simulation.

Theorem 3 *Algorithm 1 has a worst-case running time polynomial in $|\Theta_1| \times \ldots \times |\Theta_k| \times |L| \times \max_{s_i \in \Theta_i}(h^*(s_i) + \mathcal{K}) \times \gcd(\{c_l \mid l \in L\})$.*

[1] The path $u' \xrightarrow{\tau}{}^* t'$ is implicitly considered by $\mathcal{D}(s', u')$.

Algorithm 1: Quantitative LD simulation

Input: LTSs: $\mathcal{T} = \{\Theta_1, \dots, \Theta_k\}$, Limit: $\mathcal{K} \in \mathbb{N}$
Output: Dominance Function $\mathcal{D}_{\mathcal{F}} = \{\mathcal{D}_1, \dots, \mathcal{D}_k\}$
$\mathcal{D}_i[s,t] \leftarrow \max_{s_g \in S_i^G} -h^\tau(t, s_g) \; \forall t \in \Theta_i, s \in S_i^G$
$\mathcal{D}_i[s,t] \leftarrow h^*(s) - h^*(t) \; \forall t \in \Theta_i, s \notin S_i^G$
while $\exists i \in [1,k], s,t \in \Theta_i$ *s.t.* $\mathcal{D}_i[s,t] > f_{QLD}(\mathcal{T}, \mathcal{D}_{\mathcal{F}}, i, s, t)$
  if $f_{QLD}(\mathcal{T}, \mathcal{D}_{\mathcal{F}}, i, s, t) > -\mathcal{K}$ **then**
    $\mathcal{D}_i[s,t] \leftarrow f_{QLD}(\mathcal{T}, \mathcal{D}_{\mathcal{F}}, i, s, t)$
  else
    $\mathcal{D}_i[s,t] \leftarrow -h^\tau(t,s)$
return $\{\mathcal{D}_1, \dots, \mathcal{D}_k\}$

Proof Sketch: Each iteration takes polynomial time in the size of the input, i.e., the LTSs and L. At each iteration the value of some $\mathcal{D}_i(s,t)$ decreases by at least $\gcd(\{c_l \mid l \in L\})$, so the number of iterations is polynomially bounded by the number of times the number can decrease. The maximum value in the initialization is bounded by $\max_{s_i \in \Theta_i} h^*(s_i)$, and the minimum by -$\mathcal{K}$. □

In practice we set $\mathcal{K}$ to a lower value. While this diminishes the power to infer negative dominance below $-\mathcal{K}$, those are of little use anyway, since they will only be useful to prune states with very large g-value. Note that, even though the algorithm does not run in polynomial time (since $h^*(s_i)$ may be exponential in the size of the input, depending on the labels' cost), this is not a major inconvenience in practice. Other pruning techniques, like symmetry pruning (Pochter, Zohar, and Rosenschein 2011; Domshlak, Katz, and Shleyfman 2012), also rely on non-polynomial algorithms in their precomputation phase. This is not a problem, as soon as the algorithm finishes in a reasonable amount of time for tasks that are solvable without any pruning.

Advantages of Quantitative LD Simulation

Qualitative dominance pruning methods prune a node n_s if there exists another n_t s.t. $g(n_t) \leq g(n_s)$ and $s \preceq t$. An advantage of quantitative dominance is that, even when restricted to this type of pruning, QLD simulations will find coarser relations.

Theorem 4 *Let $\preceq$ and $\mathcal{D}$ be the coarsest qualitative and maximal quantitative LD simulation, respectively. Then, $\preceq \subseteq \preceq^0_{\mathcal{D}}$ and there are cases where $\preceq \subset \preceq^0_{\mathcal{D}}$.*

Proof Sketch: For $\preceq \subseteq \preceq^0_{\mathcal{D}}$. Define $\mathcal{D}(s,t) = 0$ if $s \preceq t$ and $-\infty$ otherwise. Then, $\mathcal{D}$ is an QLD simulation.

For $\preceq \subset \preceq^0_{\mathcal{D}}$, consider our example where no qualitative dominance can possibly be found for states that differ in the position of the truck. However, $T_B P_A \preceq^0_{\mathcal{D}} T_A P_T$, since $\mathcal{D}(T_A, T_B) = -h^\tau(T_B, T_A) = -1$, and $\mathcal{D}(P_A, P_T) = 1$, we can compensate the truck being at a different location if we have picked up or delivered more packages. □

Moreover, we can trade off dominance and g-value to further increase the amount of pruning.

Theorem 5 *Let $\mathcal{D}$ be a dominance function. Let n_s be a search node with state s. If there exists $n_t \in open \cup closed$ s.t. $\mathcal{D}^\epsilon(s,t) + g(n_s) - g(n_t) \geq 0$ where $\mathcal{D}^\epsilon(s,t) = \mathcal{D}(s,t) - \epsilon$ if $\mathcal{D}(s,t) < 0$ and $\mathcal{D}(s,t)$ otherwise. Then, pruning n_s preserves completeness and optimality of the algorithm.*

Proof Sketch: Since $g(n_t) + h^*(t) \leq g(n_s) + h^*(s)$, if an optimal plan from $\mathcal{I}$ to $\mathcal{G}$ goes through n_s, then $g(n_s) = g^*(s)$ and there is another optimal plan through n_t. If s is in the path from t to the goal, then $\mathcal{D}(s,t) < 0$. This means that $g(n_t) + h^*(t) + \epsilon = g^*(s) + h^*(s) + \epsilon \leq g(n_s) + h^*(s)$, so $g^*(s) < g(n_s)$, reaching a contradiction. □

Theorem 5 generalizes the qualitative pruning condition. For nodes n_s, n_t s.t. $g(n_s) = g(n_t)$ nothing changes, since n_s is pruned iff $s \preceq^0_{\mathcal{D}} t$. However, if $g(n_s) \neq g(n_t)$ we can leverage quantitative dominance to get more pruning:

- If $g(n_s) < g(n_t)$, qualitative dominance cannot prune n_s. Now, n_s may still be pruned if $\mathcal{D}(s,t)$ is high enough. This is specially relevant in A*. If there is some n_t in the closed list with a higher g-value than that of n_s, n_t was preferred by the heuristic, so there are chances of $\mathcal{D}(s,t) > 0$, assuming that dominance and the heuristic are correlated.
- If $g(n_t) < g(n_s)$, we replace the relation $\preceq^0_{\mathcal{D}}$ by the coarser $\preceq^{g(n_t)-g(n_s)+\epsilon}_{\mathcal{D}}$. This may be useful in practice because the successors of t do not necessarily dominate s or its successors according to $\preceq^0_{\mathcal{D}}$.

Action Selection Pruning

Instead of pruning states that are deemed worse than others, we may use quantitative dominance to perform action selection. Upon expansion of a node n_s, if there exists an applicable action a s.t. $s \preceq^{c(a)}_{\mathcal{D}} s[\![a]\!]$, then only that successor needs to be generated, reducing the branching factor to 1. This is safe because a starts an optimal plan from s if one exists.

Theorem 6 *Let $\mathcal{D}$ be a dominance function. Let s be a state and a an applicable action on s. If $\mathcal{D}(s, s[\![a]\!]) \geq c(a)$, then a starts an optimal plan from s to the goal if one exists.*

Proof Sketch: As $\mathcal{D}(s, s[\![a]\!]) \geq c(a)$, then $h^*(s) \geq h^*(s[a]) + c(a)$. If $c(a) > 0$, $s[\![a]\!]$ is strictly closer to the goal. If $c(a) = 0$, then $h^*(s) = h^*(t)$. By the definition of dominance function, $h^{*0}(s[\![a]\!]) \leq h^{*0}(s)$. Therefore, $s[a]$ has a path to the goal that does not go through s. □

In our running example, this is extremely powerful. Whenever a package may be loaded into the truck or unloaded at its destination this is automatically done. Since the state resulting of unloading a package in any other location is dominated by its parent, combining both types of pruning the search will only branch over driving actions.

Action selection pruning is related to other heuristic or learning methods that detect useless actions (Wehrle, Kupferschmid, and Podelski 2008) or even directly decide what action(s) to apply in certain states (Leckie and Zukerman 1998; de la Rosa et al. 2011; Krajnansky et al. 2014). Contrary to our pruning, these methods do not preserve

Blind

	#	Qualitative			Quantitative		Action Selection			POR
		$\preceq$	$\preceq_{\mathcal{D}}^{0-}$	$\preceq_{\mathcal{D}}^{0}$	$\mathcal{D}_{\overline{\tau}}$	$\mathcal{D}$	—	$\preceq_{\mathcal{D}}^{p}$	$\mathcal{D}$	
Airport(50)	15	1.3	1.3	1.3	1.3	1.3	1.1	1.1	1.3	**4.3**
Driverlog(20)	7	12.6	13.3	20.8	12.6	21.1	3.7	6.9	**27.4**	1.0
Floortile(40)	2	**140.5**	**140.5**	**140.5**	**140.5**	**140.5**	1.0	**140.5**	**140.5**	1.2
Gripper(20)	7	2.0	**2.1**	**2.1**	2.0	**2.1**	1.0	1.0	**2.1**	1.0
Logistics(63)	12	16.8	67.4	149.1	16.8	150.9	35.0	46.5	**166.0**	1.1
Maintenance(5)	5	8848.4	8848.4	35338.8	8848.4	36181.5	11617.2	46540.2	**102514.3**	3513.7
Miconic(150)	50	23.9	75.9	325.2	23.9	328.1	7.6	142.6	**376.4**	1.0
Mystery(30)	11	**1.3**	**1.3**	**1.3**	**1.3**	**1.3**	1.0	1.0	**1.3**	1.0
NoMystery(20)	8	693.3	693.3	891.5	693.3	891.5	605.8	1249.2	**10538.4**	1.1
OpenStack(100)	30	1.2	1.2	1.2	1.2	1.2	1.2	1.2	**1.3**	1.2
ParcPrint(50)	16	815.3	840.7	955.5	820.5	955.5	622.9	942.8	3542.1	**16446.1**
Path-noneg(30)	4	11.1	13.4	23.2	11.1	23.2	1.6	12.5	26.8	**29.7**
Psr-small(50)	48	**1.9**	**1.9**	**1.9**	**1.9**	**1.9**	1.7	**1.9**	**1.9**	1.2
Rovers(40)	5	29.6	93.7	396.9	29.6	396.9	62.8	203.9	**1065.8**	34.7
Satellite(36)	5	90.9	100.7	**142.8**	90.9	**142.8**	1.0	39.5	**142.8**	122.4
Scanalyzer(50)	9	1.0	1.0	1.0	1.0	**1.1**	1.0	1.0	**1.1**	1.0
Sokoban(50)	21	**1.3**	**1.3**	**1.3**	**1.3**	**1.3**	1.2	1.2	**1.3**	1.0
Tidybot(20)	1	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	**2.4**
TPP(30)	6	16.3	17.5	86.5	16.3	86.5	6.6	10.9	**102.4**	1.0
Trucks(30)	6	44.1	44.1	44.2	44.1	**44.3**	1.3	7.2	**44.3**	1.0
VisitAll(40)	12	27.8	31.1	31.1	27.8	34.9	1.0	1.0	**35.2**	1.0
Woodwork(50)	11	1626.8	1796.1	2818.2	1630.5	2820.0	480.5	2618.2	**10795.6**	549.5
Zenotravel(20)	7	5.5	8.8	21.5	5.5	21.5	2.1	2.5	**22.5**	1.1

LM-cut

	#	Qualitative			Quantitative		Action Selection			POR
		$\preceq$	$\preceq_{\mathcal{D}}^{0-}$	$\preceq_{\mathcal{D}}^{0}$	$\mathcal{D}_{\overline{\tau}}$	$\mathcal{D}$	—	$\preceq_{\mathcal{D}}^{p}$	$\mathcal{D}$	
Airport(50)	24	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Driverlog(20)	13	1.4	1.5	2.3	3.6	**4.3**	1.5	1.8	**4.3**	1.0
Floortile(40)	13	**3.5**	**3.5**	**3.5**	**3.5**	**3.5**	1.0	**3.5**	**3.5**	1.0
Gripper(20)	7	2.0	2.1	2.1	**2.8**	**2.8**	1.0	1.0	**2.8**	1.0
Logistics(63)	26	1.4	4.9	47.4	80.5	81.2	29.6	30.2	**83.9**	2.3
Maintenance(5)	5	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Miconic(150)	141	1.0	1.4	1.4	**1.7**	**1.7**	1.2	1.2	**1.7**	1.0
Mystery(30)	16	**1.2**	**1.2**	**1.2**	**1.2**	**1.2**	1.0	1.0	**1.2**	1.0
NoMystery(20)	14	4.0	4.0	4.0	45.3	45.3	16.7	18.4	**52.7**	1.0
OpenStack(100)	35	1.5	1.5	1.5	1.5	1.5	1.4	1.4	**1.9**	1.2
ParcPrint(50)	31	7.0	7.2	7.4	78.4	78.9	25.3	29.3	94.8	**1455.6**
Path-noneg(30)	5	1.7	1.9	2.7	3.3	3.4	1.3	2.0	3.4	**9.4**
Psr-small(50)	48	**1.7**	**1.7**	**1.7**	**1.7**	**1.7**	1.5	1.6	**1.7**	1.1
Rovers(40)	7	2.3	3.7	9.6	10.9	12.2	3.7	5.1	**14.8**	4.4
Satellite(36)	7	2.1	2.2	2.6	2.9	2.9	1.0	2.0	2.9	**25.7**
Scanalyzer(50)	19	1.0	1.0	1.2	**2.1**	**2.1**	1.0	1.0	**2.1**	1.0
Sokoban(50)	40	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Tidybot(20)	9	**10.3**	**10.3**	**10.3**	**10.3**	**10.3**	1.0	1.0	**10.3**	1.3
TPP(30)	6	1.8	1.8	2.5	5.9	6.7	24.7	24.7	**30.9**	1.0
Trucks(30)	10	**1.2**	**1.2**	**1.2**	**1.2**	**1.2**	1.0	1.0	**1.2**	1.0
VisitAll(40)	14	5.9	6.0	6.0	5.9	**6.1**	1.0	1.0	**6.1**	1.0
Woodwork(50)	29	5.8	5.9	8.6	52.2	52.5	2.8	2.8	76.5	**133.7**
Zenotravel(20)	13	3.9	4.0	4.7	7.3	7.3	1.3	1.5	**7.5**	1.0

Table 1: Ratio of expansions until the last f-layer by each method against the baseline in commonly solved instances (#). Domains where none of the methods obtains at least a ratio of 1.2 are excluded.

completeness and optimality. Partial-order reduction techniques like strong stubborn sets (Wehrle and Helmert 2012; Wehrle et al. 2013; Wehrle and Helmert 2014) also reduce the branching factor. However, they are based on a different notion of action interference, and indeed they do not apply in our running example because (un)load actions interfere with driving actions.

Experiments

We run experiments on all the optimal-track STRIPS planning instances from the international planning competitions (IPC'98 – IPC'14). All experiments were conducted on a cluster of Intel Xeon E5-2650v3 machines with time (memory) cut-offs of 30 minutes (4 GB). Our main objective is to compare quantitative and qualitative dominance. We run A* with the blind heuristic and LM-cut (Helmert and Domshlak 2009). We use the same initial set of LTSs for all configurations, derived by running M&S with the merge DFP strategy (Dräger, Finkbeiner, and Podelski 2006; 2009; Sievers, Wehrle, and Helmert 2014), without label reduction nor any shrinking, and with a time limit of 10 000 abstract transitions and 300 seconds. We use $\mathcal{K} = 10$.[2] These limits are adequate to finish the precomputation phase in a reasonable time (under $30s$ in most domains, though it runs out of time in a few cases). For comparison against other pruning methods, we include partial-order reduction (POR) based on strong stubborn sets (Wehrle and Helmert 2014).

Pruning power

We start by analyzing the potential of action selection (AS) and dominance pruning based on comparing each node against previously expanded states. Table 1 shows the ratio of expansions until the last f-layer of each configuration compared to the baseline without pruning. We consider multiple variants, ranging from qualitative pruning ($\preceq$) to full quantitative pruning ($\mathcal{D}$). In the middle, we consider several approximations to analyze where the gain comes from. $\preceq_{\mathcal{D}}^{0-}$ and $\preceq_{\mathcal{D}}^{0}$ perform the same pruning as $\preceq$, constructing a qualitative relation out of the quantitative dominance function. $\preceq_{\mathcal{D}}^{0-}$ defines each $\preceq_i$ as $s_i \preceq_i t_i$ iff $\mathcal{D}_i(s_i, t_i) \geq 0$ and then composes them. $\preceq_{\mathcal{D}}^{0}$ is always stronger since it trades negative dominance in one $\mathcal{D}_i$ by positive dominance in another. Quantitative dominance methods use the full strength of the quantitative function by comparing against states with different g value. $\mathcal{D}_{\overline{\tau}}$ disables τ-labels to measure their relevance.

To implement all of the above, we adapt the BDD-based method used by (Torralba and Hoffmann 2015) in which for each possible g-value they generate a BDD with all the states dominated by any state expanded with that g-value. For quantitative dominance, every time a state t is expanded, we insert the sets of states dominated by it in the corresponding $g(t) - \mathcal{D}(s, t)$ bucket. This has an important computational overhead in the qualitative case, which often becomes prohibitive with quantitative dominance. To obtain a more practical method, we use an approximation $\preceq_{\mathcal{D}}^{p}$ that prunes any state that is dominated by its parent. This greatly reduces the overhead since it ignores all previously expanded states.

Obs. 1: *Quantitative dominance is applicable in the same domains as qualitative dominance, but has a larger pruning potential.* The only exception is Scanalyzer where qualitative dominance does not achieve any pruning, but positive dominance has synergy with the LM-cut heuristic. However, among the domains where both techniques apply, quantitative dominance reduces the number of states in one or two orders of magnitude more than qualitative dominance. The gain comes from difference sources. In some domains, $\preceq_{\mathcal{D}}^{0-}$ is already stronger than $\preceq$, showing the ability of QLD simulation to find coarser relations. Trading off negative and positive dominance to construct a relation ($\preceq_{\mathcal{D}}^{0}$) already

[2] Larger values for $\mathcal{K}$ are possible, but they were not observed to significantly affect the results during our preliminar experiments.

	Blind						LM-cut					
	B	$\preceq$		AS		POR	B	$\preceq$		AS		POR
		$\preceq$	$\preceq_{TH}$	$\preceq^p_{\mathcal{D}}$	$\preceq^0_{\mathcal{D}}$			$\preceq$	$\preceq_{TH}$	$\preceq^p_{\mathcal{D}}$	$\preceq^0_{\mathcal{D}}$	
Airport(50)	**22**	15	15	**22**	15	21	28	28	28	27	26	**29**
Driverlog(20)	7	9	9	**10**	8	7	13	13	13	13	**14**	13
Elevators(50)	**26**	25	25	**26**	24	**26**	40	40	40	40	40	40
Floortile(40)	2	11	11	**16**	11	2	13	**16**	**16**	**16**	**16**	13
FreeCell(80)	**20**	**20**	**20**	**20**	**20**	14	15	15	15	15	15	15
Gripper(20)	8	8	**14**	8	8	8	7	7	**14**	7	7	7
Hiking(20)	**11**	**11**	**11**	**11**	**11**	8	9	9	9	9	9	9
Logistics(63)	12	21	20	**27**	25	12	26	26	26	**33**	28	27
Miconic(150)	55	60	61	**77**	62	50	141	141	141	**142**	141	141
Mprime(35)	**20**	19	19	**20**	19	19	22	22	22	22	22	22
Mystery(30)	**15**	11	12	**15**	11	**15**	**17**	16	**17**	**17**	**17**	**17**
NoMystery(20)	8	16	18	**20**	**20**	8	14	**20**	**20**	**20**	**20**	14
OpenStack(100)	49	51	53	55	**56**	50	47	51	48	52	**53**	49
ParcPrint(50)	16	32	31	44	28	**50**	31	35	31	48	40	**50**
Path-noneg(30)	4	4	4	**5**	4	4	5	5	5	5	5	5
PipesNT(50)	**17**	**17**	**17**	**17**	**17**	14	17	17	17	17	17	17
PipesT(50)	12	**13**	12	12	**13**	9	12	12	12	12	12	12
Psr-small(50)	**49**	**49**	**49**	48	48	**49**	**49**	**49**	**49**	48	48	**49**
Rovers(40)	6	**8**	**8**	**8**	**8**	7	7	9	9	**10**	8	**10**
Satellite(36)	6	6	6	6	6	6	7	10	10	**12**	11	**12**
Scanalyzer(50)	**21**	19	**21**	17	17	13	**27**	21	23	23	23	**27**
Sokoban(50)	41	43	**44**	43	43	39	**50**	49	48	49	49	**50**
Tetris(17)	**9**	**9**	**9**	8	8	5	**6**	**6**	5	**6**	**6**	**6**
Tidybot(40)	**16**	1	1	15	1	7	**23**	10	14	22	10	22
TPP(30)	6	6	6	6	6	6	7	7	7	**8**	**8**	6
Transport(70)	**24**	**24**	**24**	**24**	**24**	23	23	23	23	23	23	23
Trucks(30)	6	**8**	**8**	**8**	**8**	6	10	10	10	10	10	10
VisitAll(40)	12	**13**	**13**	12	**13**	12	15	**16**	**16**	15	**16**	15
Woodwork(50)	11	30	30	**38**	36	24	29	48	43	**50**	**50**	46
Zenotravel(20)	8	**9**	**9**	**9**	8	8	13	13	13	13	13	13
Others(231)	91	91	91	91	91	91	112	112	112	112	112	112
Total(1612)	610	659	671	**738**	669	613	835	856	856	**896**	869	881

Table 2: Coverage of the baseline (B), qualitative dominance, action selection (AS) with quantitative dominance, and partial-order reduction (POR).

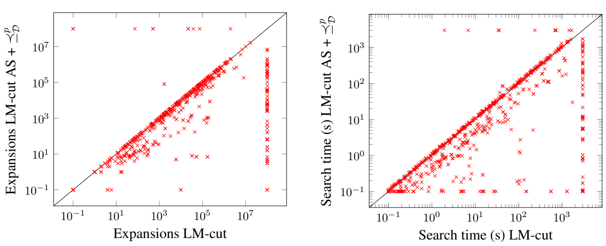


Figure 2: Expansions until last f-layer and search time of AS + $\preceq^p_{\mathcal{D}}$ against the baseline with LM-cut.

achieves most of the pruning in several domains, specially in blind search. Trading off dominance and g-value ($\mathcal{D}$) is more relevant with heuristics (e.g., NoMystery). The potential of quantitative dominance is also reflected in the comparison against POR, since it is able to achieve stronger pruning in most domains. Finally, the consideration of τ labels can be seen important in around half of the domains, sometimes increasing the pruning in one order of magnitude.

Obs. 2: *Action selection pruning is highly complementary to previous dominance pruning methods.* In most domains, the combination of both methods is stronger than any of them. Moreover, since the overhead of action selection is quite low, it is almost always worth to use it whenever a quantitative dominance function has been computed.

Overall Performance

Table 2 compares the coverage of our two best methods, AS with pruning against the parent or against previously expanded nodes, against qualitative dominance and POR. For a fair comparison, we include qualitative pruning with the same input LTSs as our approach ($\mathcal{D}$) and the configuration used by Torralba and Hoffmann(2015) ($\preceq_{TH}$) which uses exact label reduction (Sievers, Wehrle, and Helmert 2014), bisimulation shrinking (Nissim, Hoffmann, and Helmert 2011) and a larger LTS size (100k). All configurations except $\preceq^p_{\mathcal{D}}$ use the "safety belt" that disables the method if no pruning has been achieved after 1000 expansions.

Obs. 3: *AS + $\preceq^p_{\mathcal{D}}$ has huge pruning power and low overhead, greatly increasing the capabilities of heuristic search planners.* It obtains the best overall coverage, solving 128 instances over the baseline in blind search and 61 with LM-cut, much higher than POR or qualitative dominance. Some domains like NoMystery that are hard even when using good heuristics, become simple under the analysis of quantitative dominance, which even with blind search is able to solve all tasks. Figure 2 directly compares the number of expanded nodes and search time of AS + $\preceq^p_{\mathcal{D}}$ against the baseline. It obtains reductions of several orders of magnitude in the number of expansions with little overhead. Note that this ignores the precomputation time (which can be of up to 300s to compute the LTSs plus the computation of the QLD simulation), but, as the coverage improvement shows, the precomputation time is highly compensated by the search space reduction in instances that are not quickly solved by the baseline.

Obs. 4: *The overhead of current methods for exploiting the full potential of quantitative dominance ($\mathcal{D}$) is too high to pay off.* The $\mathcal{D}$ configuration did not improve the other methods anywhere and was excluded from the table. This contrasts with the results of Table 1 that show a great potential. However, there are a few domains where the additional pruning when using $\preceq^0_{\mathcal{D}}$ to complement AS pays off like Driverlog, Openstacks or VisitAll. Further exploring this trade-off between pruning power and overhead (e.g., using dominance-based methods for irrelevance pruning (Torralba and Kissmann 2015)) is an interesting topic for future work.

Conclusion

We have introduced the notion of quantitative dominance for optimal planning, which extends previous approaches of qualitative dominance. This extension is more effective at analyzing the structure of the task, which leads to stronger pruning. More importantly, the quantitative information enables new ways of pruning. We introduced action selection pruning, a novel pruning method that applies a single action on a state if the action starts an optimal plan from the state according to the quantitative dominance function. Our experiments show that action selection is highly complementary to previous dominance pruning methods, greatly extending the capabilities of heuristic search planners.

Proofs

In this section we provide a detailed proof of Theorem 2. First, Lemma 1 shows that the property holds if there is only a single LTS $\mathcal{T} = \{\Theta\}$.

Lemma 1 *Let $\mathcal{D}$ be a goal-respecting function for Θ such that for all $s, t \in \Theta$, $\mathcal{D}(s,t) \leq min_{s \xrightarrow{l} s' \in \Theta} \max_{u \xrightarrow{l'} u' \in \Theta} \mathcal{D}(s',u') - h^\tau(t,u) + c(l) - c(l')$. Then, $\mathcal{D}$ is a quantitative dominance function for Θ.*

Proof: If $s \in S^G$, then $\mathcal{D}(s,t) \leq \max_{s_g \in S^G} -h^\tau(t, s_g) \leq -h^*(t)$. So, $\mathcal{D}(s,t) \leq h^*(s) - h^*(t) = 0 - h^*(t)$. Note that if $h^*(t) = 0$ but $t \notin S^G$ then $h^\tau(t, s_g) \geq \epsilon$ so $\mathcal{D}(s,t) < 0$.

If $s \notin S^G$, we use induction on plan length. Let $s \xrightarrow{l} s'$ be the first action in a shortest optimal plan for s. Then, there exists a path $t \xrightarrow{\tau}{}^{*} u \xrightarrow{l} u'$ s.t. $\mathcal{D}(s,t) \leq \mathcal{D}_i(s',u') - h^\tau(t,u) + c(l) - c(l')$. By induction $\mathcal{D}(s',u') \leq h^*(s') - h^*(u')$. So, $\mathcal{D}(s,t) \leq (c(l) + h^*(s')) - (h^*(u') + c(l') + h^\tau(t,u) \leq h^*(s) - h^*(t)$. □

Next, Lemmas 2 to 5 prove certain properties that will be needed for proving that the property of being an QLD simulation is invariant under the synchronized product operation.

Lemma 2 *Let $\{\mathcal{D}_1, \ldots \mathcal{D}_k\}$ be the maximal QLD simulation on $\mathcal{T} = \{\Theta_1, \ldots \Theta_k\}$. Then, for any $s_i, t_i \in \Theta_i$ and $s_j, t_j \in \Theta_j$, there exist states $(s_i s_j), (t_i t_j) \in \Theta_i \otimes \Theta_j$ and $h^\tau(s_i s_j, t_i t_j) \leq h^\tau(s_i, t_i) + h^\tau(s_j, t_j)$.*

Proof: Let L_i^τ and L_j^τ be the set of τ-labels for Θ_i and Θ_j respectively, and $L_{i,j}^\tau$ the set of τ labels for $\Theta_i \otimes \Theta_j$. Then, $L_i^\tau \cup L_j^\tau \subseteq L_{i,j}^\tau$ because all labels in L_i^τ and L_j^τ do not affect any LTS in $\mathcal{T} \setminus \{\Theta_i, \Theta_j\}$. Therefore, for any paths of τ-labels $s_i \xrightarrow{\pi_i} t_i$ and $s_j \xrightarrow{\pi_j} t_j$, we have a path $(s_i s_j) \xrightarrow{\pi_1} (t_i s_j) \xrightarrow{\pi_2} (t_i t_j)$. □

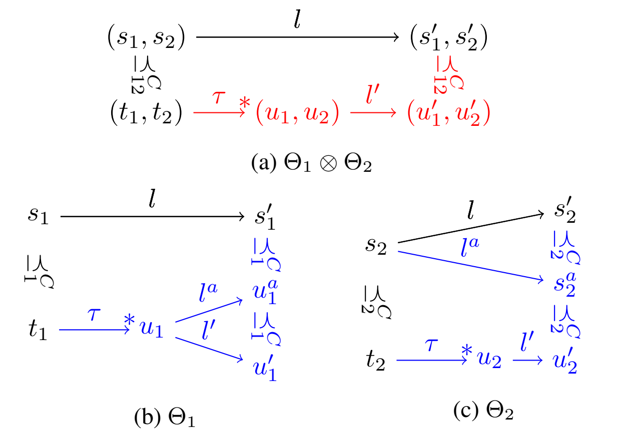
Lemma 3 *Let $\{\mathcal{D}_1, \ldots \mathcal{D}_k\}$ be the maximal QLD simulation on $\{\Theta_1, \ldots \Theta_k\}$. Then, for all $i \in [1,k]$:*

(i) $\forall s, t, u \in \Theta_i$ $\mathcal{D}_i(s,t) + \mathcal{D}_i(t,u) \leq \mathcal{D}_i(s,u)$

(ii) $\forall l_1, l_2, l_3 \in L$ $\mathcal{D}_i^L(l_1, l_2) + \mathcal{D}_i^L(l_2, l_3) \leq \mathcal{D}_i^L(l_1, l_3)$.

Proof: First we prove (i) by contradiction. Assume that there exist states s, t, u s.t. $\mathcal{D}_i(s,t) + \mathcal{D}_i(t,u) > \mathcal{D}_i(s,u)$. Then, define $\mathcal{D}$' by setting $\mathcal{D}' = \mathcal{D}$ and iteratively assigning $\mathcal{D}_i'(s,u) = \mathcal{D}_i(s,t) + \mathcal{D}_i(t,u)$ for any s,t,u s.t. $\mathcal{D}_i(s,t) + \mathcal{D}_i(t,u) > \mathcal{D}_i(s,u)$ until a fixpoint is reached.

Then, $\mathcal{D}$' is also an QLD simulation. Increasing $\mathcal{D}_i(s,u)$ can only cause the values of $\mathcal{D}^L$ and $f_{QLD}$to increase. Therefore, the inequality $\mathcal{D}_i(x,y) \leq f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, x, y)$ still holds for any $\langle x, y\rangle \neq \langle s, u\rangle$. The inequality $\mathcal{D}_i'(s,u) \leq f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, s, u)$ also holds because for any $s \xrightarrow{l} s'$ there exists a path $t \xrightarrow{\tau}{}^{*} t^a \xrightarrow{l'} t'$ s.t. $\mathcal{D}(s,t) \leq \mathcal{D}_i(s',t') - h^\tau(t,t^a) + c(l) - c(l') + \sum_{j \neq i} \mathcal{D}_j^L(l,l')$. Same for $\mathcal{D}(t,u)$ where there exists a path $u \xrightarrow{\tau}{}^{*} u^a \xrightarrow{l''} u'$ s.t. $\mathcal{D}(t,u) \leq \mathcal{D}_i(t',u') - h^\tau(t,t^a) + c(l') - c(l'') + \sum_{j \neq i} \mathcal{D}_j^L(l', l'')$. Adding both inequalities we obtain $\mathcal{D}'(s,u) \leq f_{QLD}(\mathcal{T}, \mathcal{D}_\mathcal{F}, i, s, u)$. Therefore, $\mathcal{D}$ was not the maximal function satisfying this property.



(a) $\Theta_1 \otimes \Theta_2$

(b) Θ_1

(c) Θ_2

Figure 3: Illustration for the proof of Theorem 2. We use a color code to highlight what holds by assumption and the definition of synchronized product (in black), what we need to prove (red), and the intermediate deduction steps (blue).

Claim (ii) follows from (i):

$$\mathcal{D}_i^L(l_1, l_2) + \mathcal{D}_i^L(l_2, l_3) =$$
$$\min_{s \xrightarrow{l_1} t} \max_{s \xrightarrow{l_2} u} \mathcal{D}_i(t,u) + \min_{s \xrightarrow{l_2} u} \max_{s \xrightarrow{l_3} v} \mathcal{D}_i(u,v) \leq$$
$$\min_{s \xrightarrow{l_1} t} \max_{s \xrightarrow{l_2} u} \mathcal{D}_i(t,u) + \max_{s \xrightarrow{l_3} v} \mathcal{D}_i(u,v) \leq$$
$$\min_{s \xrightarrow{l_1} t} \max_{s \xrightarrow{l_3} v} \mathcal{D}_i(t,v) = \mathcal{D}_i^L(l_1, l_3)$$

□

Lemma 4 *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two goal-respecting functions for Θ_1 and Θ_2, respectively. Then, $\mathcal{D}_1 + \mathcal{D}_2$ is a goal-respecting function for $\Theta_1 \otimes \Theta_2$.*

Proof: Consider any goal state (s_1, s_2) and non-goal state (t_1, t_2) in $\Theta_1 \otimes \Theta_2$. By the definition of synchronized product, s_1 and s_2 are goal states in Θ_1 and Θ_2, respectively. Therefore, $\mathcal{D}_1(s_1, t_1) + \mathcal{D}_2(s_2, t_2) \leq \max_{s_1^g \in S_1^G} -h^\tau(t_1, s_1^g) + \max_{s_2^g \in S_2^G} -h^\tau(t_2, s_2^g)$. By Lemma 2, this cannot be greater than $\max_{(s_1^g, s_2^g) \in S^G} -h^\tau((t_1, t_2), (s_1^g, s_2^g))$. □

Lemma 5 *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two functions for Θ_1 and Θ_2, respectively, and $\mathcal{D}_{1,2} := \mathcal{D}_1 + \mathcal{D}_2$ be a function for $\Theta_1 \otimes \Theta_2$. Then, $\mathcal{D}_1^L(l,l') + \mathcal{D}_2^L(l,l') \leq \mathcal{D}_{1,2}^L(l,l')$.*

Proof: Let $s \xrightarrow{l} s' \in \Theta_{1,2}$ be the transition that minimizes the value of $\mathcal{D}_{1,2}^L(l,l')$. Then, there exist $s_1 \xrightarrow{l} s_1' \in \Theta_1$ and $s_2 \xrightarrow{l} s_2' \in \Theta_2$. So, there exists $s_1 \xrightarrow{l'} t_1 \in \Theta_1$ s.t. $\mathcal{D}_1^L(l,l') \leq \mathcal{D}_1(s_1', t_1)$ and analogously for Θ_2. Therefore, there exists $t = (t_1, t_2) \in \Theta_{1,2}$ s.t. $s \xrightarrow{l'} t'$ and the following inequality holds:

$$\mathcal{D}_1^L(l,l') + \mathcal{D}_2^L(l,l') \leq$$
$$\mathcal{D}_1(s_1', t_1) + \mathcal{D}_2(s_2', t_2) = \mathcal{D}_{1,2}(s', t) \leq \mathcal{D}_{1,2}^L(l,l')$$

□

$$\mathcal{D}_1(s_1,t_1) \le \mathcal{D}_1(s_1',u_1^a) - h^\tau(t_1,t_1') + c(l) - c(l^a) + \sum_{j\in 2,\dots,k} \mathcal{D}_j^L(l,l^a) \quad (1)$$

$$\mathcal{D}_2^L(l,l^a) \le \mathcal{D}_2(s_2',s_2^a) \quad (2)$$

$$\mathcal{D}_2(s_2,t_2) \le \mathcal{D}_2(s_2^a,u_2') - h^\tau(t_1,t_1') + c(l^a) - c(l') + \sum_{j\in 2,\dots,k} \mathcal{D}_j^L(l^a,l') \quad (3)$$

$$\mathcal{D}_1^L(l^a,l') \le \mathcal{D}_1(u^a,u_1') \quad (4)$$

$$\forall \Theta_i \in \{\Theta_1,\dots,\Theta_k\}\ \forall_{s,t,u\in\Theta_i}\ \mathcal{D}_i(s,t) + \mathcal{D}_i(t,u) \le \mathcal{D}_i(s,u) \quad (5)$$

$$\forall \Theta_i \in \{\Theta_1,\dots,\Theta_k\} \forall_{l,l',l''}\ \mathcal{D}_i^L(l,l') + \mathcal{D}_i^L(l',l'') \le \mathcal{D}_i^L(l,l'') \quad (6)$$

$$h^\tau((s_1,s_2),(t_1,t_2)) \le h^\tau(s_1,t_1) + h^\tau(s_2,t_2) \quad (7)$$

$$\mathcal{D}_{1,2}((s_1,s_2),(t_1,t_2)) = \mathcal{D}_1(s_1,t_1) + \mathcal{D}_2(s_2,t_2) \quad (8)$$

$$\begin{aligned}
&\mathcal{D}_{1,2}(s,t) = \mathcal{D}_1(s_1,t_1) + \mathcal{D}_2(s_2,t_2)\\
\text{(by 1, 3)} &\le \mathcal{D}_1(s_1',u_1^a) + c(l) - c(l^a) - h^\tau(t_1,u_1) + \sum_{j\neq 1} \mathcal{D}_j^L(l,l^a) + \mathcal{D}_2(s_2^a,u_2') + c(l^a) - c(l') - h^\tau(t_2,u_2) + \sum_{j\neq 2} \mathcal{D}_j^L(l^a,l')\\
\text{(by 2, 4)} &\le \mathcal{D}_1(s_1',u_1^a) + \mathcal{D}_1(u_1^a,u_1') + \mathcal{D}_2(s_2',s_2^a) + \mathcal{D}_2(s_2^a,u_2') + c(l) - c(l') - h^\tau(t_1,u_1) - h^\tau(t_2,u_2) + \sum_{j>2} (\mathcal{D}_j^L(l,l^a) + \mathcal{D}_j^L(l^a,l'))\\
\text{(by 5, 6)} &\le \mathcal{D}_1(s_1',u_1') + \mathcal{D}_2(s_2',u_2') + c(l) - c(l') - h^\tau(t_1,u_1) - h^\tau(t_2,u_2) + \sum_{j>2} \mathcal{D}_j^L(l,l')\\
\text{(by 7, 8)} &\le \mathcal{D}_{1,2}((s_1',s_2'),(u_1',u_2')) + c(l) - c(l') - h^\tau((t_1,t_2),(u_1',u_2')) + \sum_{j>2} \mathcal{D}_j^L(l,l')
\end{aligned}$$

Figure 4: Derivation of the inequality required by the proof of Theorem 2. Inequalities that have already been proven (above) and how are they applied to reach the desired inequality (below).

Theorem 2 *Let $\mathcal{D}_\mathcal{F} = \{\mathcal{D}_1,\dots,\mathcal{D}_k\}$ be an QLD simulation on $\mathcal{T} = \{\Theta_1,\dots,\Theta_k\}$. Then, $\mathcal{D}_1 + \dots + \mathcal{D}_k$ is a quantitative dominance function on $\Theta_1 \otimes \dots \otimes \Theta_k$.*

Proof: We assume that $\mathcal{D}_\mathcal{F}$ is the maximal QLD simulation. Note that if it is a dominance function any other QLD simulation must be as well because decreasing the values of the function cannot possibly cause the condition of dominance function to become false.

If there is a single LTS, the proof follows from Lemma 1. Next we show that the property of being an QLD simulation is invariant under the synchronized product operation. Assume WLOG that we merge Θ_1 and Θ_2 to obtain $\mathcal{T}' = \{\Theta_1 \otimes \Theta_2, \Theta_3,\dots,\Theta_k\}$, and $\mathcal{D}'_\mathcal{F} = \{\mathcal{D}_{1,2},\mathcal{D}_3,\dots,\mathcal{D}_k\}$ where $\mathcal{D}_{1,2} = \mathcal{D}_1 + \mathcal{D}_2$. In the following we show that $\mathcal{D}'_\mathcal{F}$ is an QLD simulation for $\mathcal{T}'$.

Lemma 4 ensures that $\mathcal{D}_{1,2}$ is goal-respecting. Next, we show that the $\mathcal{D}_i(s,t) \le f_{QLD}(\mathcal{T}',\mathcal{D}'_\mathcal{F},i,s,t)$ holds after merging Θ_1 and Θ_2. First, let's consider the case of other LTSs where $i > 2$. In order for the inequality $\mathcal{D}_i(s,t) \le f_{QLD}(\mathcal{T}',\mathcal{D}'_\mathcal{F},i,s,t)$ to be preserved, the values of $\mathcal{D}^L$ must not decrease after merging Θ_1 and Θ_2, i.e., $\mathcal{D}_1^L(l,l') + \mathcal{D}_2^L(l,l') \le \mathcal{D}_{1,2}^L(l,l')$. This is ensured by Lemma 5.

Figure 3 illustrates the main case, where $i = (1,2)$, $s = (s_1,s_2)$, and $t = (t_1,t_2)$. The inequality holds automatically if $\mathcal{D}_1(s_1,t_1) = -\infty$ or $\mathcal{D}_2(s_2,t_2) = -\infty$ so we may assume that $s_1 \preceq^C_{\mathcal{D}1} t_1$ and $s_2 \preceq^C_{\mathcal{D}2} t_2$. We need to show that for any transition $s = (s_1,s_2) \xrightarrow{l} (s_1',s_2') = s'$, there exists a transition $(u_1,u_2) \xrightarrow{l'} (u_1',u_2')$ s.t. $\mathcal{D}_{1,2}(s,t) \le \mathcal{D}_{1,2}(s',u') + c(l) - h^\tau(t,u) - c(l') + \sum_{j\in 3,\dots,k} \mathcal{D}_j^L(l,l')$. We first prove the existence of such s', u, and u' states and then we show that the inequality holds.

In Θ_1, since $s_1 \preceq^C_{\mathcal{D}1} t_1$, there must exist $u_1 \xrightarrow{l^a} u_1^a$ s.t. (E1) $\mathcal{D}_1(s_1,t_1) \le \mathcal{D}_1(s_1',u_1^a) - h^\tau(t_1,u_1) + c(l) - c(l^a) + \sum_{j\in 2,\dots,k} \mathcal{D}_j^L(l,l^a)$. This implies that $l \preceq^L_2 l^a$. In Θ_2, since $l \preceq^L_2 l^a$ and $s_2 \xrightarrow{l} s_2'$, there must exist $s_2 \xrightarrow{l^a} s_2^a$ s.t. (E2) $\mathcal{D}_2^L(l,l^a) \le \mathcal{D}_2(s_2',s_2^a)$. Now, since $s_2 \preceq^C_{\mathcal{D}2} t_2$ there must exist $u_2 \xrightarrow{l'} u_2'$ s.t. (E3) $\mathcal{D}_2(s_2,t_2) \le \mathcal{D}_2(s_2^a,u_2') - h^\tau(t_1,u_1) + c(l^a) - c(l') + \sum_{j\in 2,\dots,k} \mathcal{D}_j^L(l^a,l')$. This implies that $l^a \preceq^L_1 l$. Going back to Θ^1, since $l^a \preceq^L_1 l$, there must exist $u_1 \xrightarrow{l'} u_1'$ s.t. (E4) $\mathcal{D}_1^L(l^a,l') \le \mathcal{D}_1(u^a,u_1')$.

To prove that the inequality holds $\mathcal{D}_{1,2}(s,t) = \mathcal{D}_1(s_1,t_1) + \mathcal{D}_2(s_2,t_2) \le \mathcal{D}_{1,2}(s',u') + c(l) - c(l') - h^\tau(t,u) + \sum_{j=3,\dots,k} \mathcal{D}_j^L(l,l')$, we substitute in the left parts the inequalities (E1-E4), the results of Lemmas 3 (E5 and E6) and 2 (E7) and the fact that $\mathcal{D}_{1,2}$ is defined as the sum of $\mathcal{D}_1$ and $\mathcal{D}_2$ (E8). Figure 4 shows all these equations and the substitutions in a step by step manner.

□

Acknowledgments

Work supported by the German Federal Ministry of Education and Research (BMBF) CISPA, grant no. 16KIS0656. Thanks to Rosa Moreno and Daniel Gnad for helpful discussions concerning this work.

References

de la Rosa, T.; Celorrio, S. J.; Fuentetaja, R.; and Borrajo, D. 2011. Scaling up heuristic planning with relational decision trees. *Journal of Artificial Intelligence Research* 40:767–813.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2012. Enhanced symmetry breaking in cost-optimal planning as forward search. In Bonet, B.; McCluskey, L.; Silva, J. R.; and Williams, B., eds., *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*. AAAI Press.

Dräger, K.; Finkbeiner, B.; and Podelski, A. 2006. Directed model checking with distance-preserving abstractions. In Valmari, A., ed., *Proceedings of the 13th International SPIN Workshop (SPIN 2006)*, volume 3925 of *Lecture Notes in Computer Science*, 19–34. Springer-Verlag.

Dräger, K.; Finkbeiner, B.; and Podelski, A. 2009. Directed model checking with distance-preserving abstractions. *International Journal on Software Tools for Technology Transfer* 11(1):27–37.

Hall, D.; Cohen, A.; Burkett, D.; and Klein, D. 2013. Faster optimal planning with partial-order pruning. In Borrajo, D.; Fratini, S.; Kambhampati, S.; and Oddi, A., eds., *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*. Rome, Italy: AAAI Press.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In Gerevini, A.; Howe, A.; Cesta, A.; and Refanidis, I., eds., *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*, 162–169. AAAI Press.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge & shrink abstraction: A method for generating lower bounds in factored state spaces. *Journal of the Association for Computing Machinery* 61(3).

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In Boddy, M.; Fox, M.; and Thiebaux, S., eds., *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, 176–183. Providence, Rhode Island, USA: Morgan Kaufmann.

Hennessy, M., and Milner, R. 1985. Algebraic laws for nondeterminism and concurrency. *Journal of the Association for Computing Machinery* 32(1):137–161.

Krajnansky, M.; Buffet, O.; Hoffmann, J.; and Fern, A. 2014. Learning pruning rules for heuristic search planning. In Schaub, T., ed., *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, 483–488. Prague, Czech Republic: IOS Press.

Leckie, C., and Zukerman, I. 1998. Inductive learning of search control rules for planning. *Artificial Intelligence* 101(1–2):63–98.

Nissim, R.; Hoffmann, J.; and Helmert, M. 2011. Computing perfect heuristics in polynomial time: On bisimulation and merge-and-shrink abstraction in optimal planning. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, 1983–1990. AAAI Press/IJCAI.

Pochter, N.; Zohar, A.; and Rosenschein, J. S. 2011. Exploiting problem symmetries in state-based planners. In Burgard, W., and Roth, D., eds., *Proceedings of the 25th National Conference of the American Association for Artificial Intelligence (AAAI'11)*. San Francisco, CA, USA: AAAI Press.

Radzi, M. 2011. *Multi-objective planning using linear programming*. Ph.D. Dissertation, University of Strathclyde.

Sievers, S.; Wehrle, M.; and Helmert, M. 2014. Generalized label reduction for merge-and-shrink heuristics. In Brodley, C. E., and Stone, P., eds., *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI'14)*, 2358–2366. Austin, Texas, USA: AAAI Press.

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

Torralba, Á., and Kissmann, P. 2015. Focusing on what really matters: Irrelevance pruning in merge-and-shrink. In Lelis, L., and Stern, R., eds., *Proceedings of the 8th Annual Symposium on Combinatorial Search (SOCS'15)*, 122–130. AAAI Press.

Wehrle, M., and Helmert, M. 2012. About partial order reduction in planning and computer aided verification. In Bonet, B.; McCluskey, L.; Silva, J. R.; and Williams, B., eds., *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12)*. AAAI Press.

Wehrle, M., and Helmert, M. 2014. Efficient stubborn sets: Generalized algorithms and selection strategies. In Chien, S.; Do, M.; Fern, A.; and Ruml, W., eds., *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*. AAAI Press.

Wehrle, M.; Helmert, M.; Alkhazraji, Y.; and Mattmüller, R. 2013. The relative pruning power of strong stubborn sets and expansion core. In Borrajo, D.; Fratini, S.; Kambhampati, S.; and Oddi, A., eds., *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS'13)*. Rome, Italy: AAAI Press.

Wehrle, M.; Kupferschmid, S.; and Podelski, A. 2008. Useless actions are useful. In Rintanen, J.; Nebel, B.; Beck, J. C.; and Hansen, E., eds., *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, September 14-18, 2008*, 388–395. AAAI Press.