Symmetry Breaking: Satisficing Planning and Landmark Heuristics

Carmel Domshlak
Technion
Haifa, Israel
dcarmel@ie.technion.ac.il

Michael Katz
Saarland University
Saarbrücken, Germany
katz@cs.uni-saarland.de

Alexander Shleyfman
Technion
Haifa, Israel
shleyfman.alexander@gmail.com

Abstract

Searching for computational tools that can further push the boundary of satisficing planning, we show that reasoning about state-space symmetries can substantially improve even the most effective heuristic-search satisficing planners, with respect to all standard performance measures. The improvement comes from the state-space pruning, as well as from transparent cost-to-state updates and heuristic enhancement by information obtained during the search at different symmetric states.

Introduction

Over the last two decades, the combined machinery of relaxation heuristics, preferred operators, and various enhancements of the very search infrastructure, have positioned heuristic forward search as a leading technique for satisficing planning, in terms of both efficiency and robustness. A prominent example is LAMA-11 (or LAMA, for short), a heuristic-search planning system that won the sequential satisficing track of the International Planning Competition (IPC) in 2011 (Richter, Westphal, and Helmert 2011), with its predecessor, LAMA-08, winning the respective IPC track in 2008. LAMA builds on the Fast Downward system (Helmert 2006), inheriting the general structure of Fast Downward, the translation of propositional PDDL tasks to representations with finite-domain variables, and the exploitation of several heuristics simultaneously via a multi-queue search architecture. The two core features of LAMA are its iterated search using restarts (Richter, Thayer, and Ruml 2010), and the use of relaxation landmarks for defining heuristic estimates and preferred operators (Richter, Helmert, and Westphal 2008).

State-of-the-art planners these days carefully balance between search completeness and focus, between the informativeness of the heuristics and the cost of computing them. A very interesting question is what additional techniques can further stratify satisficing heuristic-search planning, either in terms of coverage or in terms of plan quality, or both? While this question is broad enough to have many positive answers, with the years it is getting harder and harder to push the boundary of satisficing planning. Here we investigate prospects of reasoning about state-space symmetries within satisficing heuristic-search planning. Adopting the framework of goal-stable automorphisms for cost-optimal planning with A^* (Domshlak, Katz, and Shleyfman 2012), we show that its simple adaptation to greedier search procedures results in substantial state pruning, as well as in transparent improvement of discovered plan quality. Furthermore, we show that goal-stable automorphism groups such as those of Domshlak, Katz, and Shleyfman (2012) can be used to improve informativeness of landmark heuristic estimates, by aggregating information obtained during the search at different symmetric states. We show that both these features are very cost-effective, in the sense of robust improvement of both standard *GBFS* and LAMA's iterative search, with respect to all standard performance measures.

Background

We consider planning tasks $\Pi = \langle V, O, s_0, G, cost\rangle$ captured by the standard SAS$^+$ formalism (Bäckström and Klein 1991; Bäckström and Nebel 1995) with operator costs. V is a set of finite-domain state variables, $S = \prod_{v \in V} dom(v)$ is the state space of Π, s_0 is an initial state, and goal G is a partial assignment to V; a state s is a goal state, denoted by $s \in S_*$, iff $G \subseteq s$. O is a finite set of operators, each given by a pair $\langle \mathsf{pre}, \mathsf{eff}\rangle$ of partial assignments to V, called preconditions and effects, and $cost : O \to \mathbb{R}^{0+}$ is an operator cost function. Applying operator o in state s results in a state denoted by $s[\![o]\!]$. By the transition graph $\mathcal{T}_\Pi = \langle S, E\rangle$ of Π we refer to the edge-labeled digraph induced by Π over S: if $o \in O$ is applicable in state s, then $\mathcal{T}_\Pi$ contains an edge $(s, s[\![o]\!]; o)$ from s to $s[\![o]\!]$, labeled with o. For a task $\Pi = \langle V, O, s_0, G, cost\rangle$ and state $s \in S$, task $\Pi(s)$ is obtained from Π by setting the initial state to be s. Auxiliary notation: for $k \in \mathbb{N}$, $i \in [k]$ stands for $i \in \{1, 2, \ldots, k\}$.

The LAMA Planning System: Richter and Westphal (2010) provide a detailed description of LAMA, and thus here we briefly describe only the components relevant to our presentation later on. Using *GBFS* and then *WA**, LAMA employs two heuristics, each inducing its sets of preferred operators: the delete-relaxation FF heuristic (Hoffmann and Nebel 2001) and the landmark heuristic. The latter is based on disjunctive landmarks of the planning task, that is, sets of variable assignments of which one must occur at some point.

1. Offline: Find an equivalence relation $\sim \leq \sim_{\Gamma_{S_*}}$.
2. When evaluating the state s with $s \sim s'$ for some previously evaluated state s', if $g(s) \geq g(s')$, prune s as if it were never generated. Otherwise set g, parent, and act of s' to those of s. If WA^*, reopen s'.
3. If a goal state s_* is reached, (i) extract a sequence $\pi = \langle(\varepsilon, s_0), (o_1, s_1), \ldots, (o_m, s_m)\rangle$ of pairs of state and action, where $s_m = s_*$, by the standard backchaining from s_* along the parent relation, setting actions by the act relation, and (ii) return trace-forward(π).

Figure 1: *GBFS*/*WA** extension to Γ_{S_*} symmetry breaking

Given a state s and a set L of Π's landmarks, possibly annotated with some orderings, the *landmark heuristic* estimate of s is set to the *number of landmarks* $L(s)$ *yet to be achieved from* s *onwards* (Richter, Helmert, and Westphal 2008). When forward search reaches s for the first time via a sequence of operators π, $L(s)$ is set to $L \setminus (\mathsf{A}(s,\pi) \setminus \mathsf{RA}(s,\pi))$, where $\mathsf{A}(s,\pi) \subseteq L$ and $\mathsf{RA}(s,\pi) \subseteq \mathsf{A}(s,\pi)$ are the sets of *accepted* and *required again* landmarks, respectively. A landmark is accepted if it occurs at some state along π; the set $\mathsf{A}(s,\pi)$ is memorized as state property $\mathsf{A}(s)$. An accepted landmark is required again if it does not hold in s and it is a direct precondition of some landmark which is not accepted. From this point on, each time s is reached via this or another operator sequence π', LAMA performs a "multi-path" revision of the landmarks accepted at s by updating $\mathsf{A}(s)$ to $\mathsf{A}(s) \cap \mathsf{A}(s,\pi)$, and then recomputing $L(s)$ as above (Karpas and Domshlak 2009).

A^* Symmetry Breaking with Γ_{S_*}: An *automorphism* of a transition graph $\mathcal{T}_\Pi = \langle S, E\rangle$ is a permutation σ of the vertices S such that $(s, s'; o) \in E$ iff, for some o' with $cost(o') = cost(o)$, $(\sigma(s), \sigma(s'); o') \in E$. Automorphisms are closed under composition, forming the *automorphism group* $Aut(\mathcal{T}_\Pi)$ of the graph. $\Gamma \leq \Gamma'$ denotes that Γ is a subgroup of Γ'. Each subgroup of automorphisms $\Gamma \leq Aut(\mathcal{T}_\Pi)$ induces an equivalence relation $\sim_\Gamma$ on states S: $s \sim_\Gamma s'$ iff $\sigma(s) = s'$ for some $\sigma \in \Gamma$. For a state subset $S' \subseteq S$, the subgroup $\Gamma_{S'} = \{\sigma \in \Gamma \mid \forall s \in S' : \sigma(s) \in S'\} \leq \Gamma$ is the *stabilizer* of $S_1, \ldots, S_k$ with respect to Γ. Finally, a set of automorphisms Σ is said to *generate* a group Γ if Γ is the fixpoint of iterative composition of the elements of Σ. Finding a generating set of $Aut(G)$ for a graph G is not known to be polynomial-time, but backtracking search techniques are surprisingly effective in finding generating sets for substantial subgroups of $Aut(G)$.

Pruning symmetries by reasoning about automorphisms of the search space has been adopted in model checking (Emerson and Sistla 2011), constraint satisfaction (Puget 1993), and planning (Rintanen 2003; Fox and Long 1999; 2002; Pochter, Zohar, and Rosenschein 2011; Domshlak, Katz, and Shleyfman 2012). Here we build upon the recent approach of Domshlak, Katz, and Shleyfman (2012) for exploiting state space symmetries in cost-optimal planning using A^*, referred to here for brevity as DKS.

At the focus of DKS is a property of plans and goal-stabilizing automorphisms Γ_{S_*}: Let Π be a planning task, $\Gamma \leq \Gamma_{S_*}$, and $(s_0, s_1, \ldots, s_k)$, $(s_0, s'_1, \ldots, s'_l)$ be a pair of

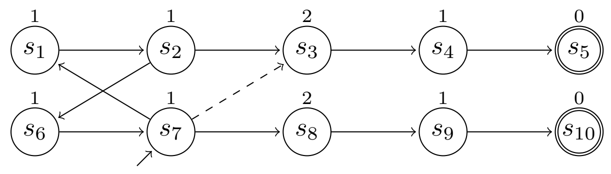

Figure 2: Illustration for Proposition 1

plans for Π. If, for some $i \in [k]$ and $i < j \in [l]$, $s_i = \sigma(s'_j)$ for some $\sigma \in \Gamma$, then $(s_0, \ldots, s_{i-1}, \sigma(s'_j), \ldots, \sigma(s'_l))$ is also a plan for Π, shorter than $(s_0, s'_1, \ldots, s'_l)$. Based on that, DKS extends A^* search as follows: No matter which of the two states s_i and s'_j as above is generated second, it is pruned from the search. However, if s_i is the state generated second, then s'_j ceases represent itself and starts representing its Γ_{S_*}-symmetric counterpart s_i. For that "role switching" of s'_j, the parent s_{i-1} of s_i "adopts" s'_j as a pseudo-child and the operator o such that $s_i = s_{i-1}[\![o]\!]$ is memorized. These "state adoptions" then should be taken into account at plan extraction; for the respective procedure, we refer the reader to Domshlak, Katz, and Shleyfman (2012).

As the transition graph $\mathcal{T}_\Pi$ is not given explicitly, automorphisms of $\mathcal{T}_\Pi$ must be inferred from the description of Π. Following Pochter, Zohar, and Rosenschein (2011), the implementation of DKS by Domshlak, Katz, and Shleyfman (2012) is restricted to certain "syntactic" automorphisms $\Gamma^{\mathrm{pdg}}_{S_*} \leq \Gamma_{S_*}$, corresponding to automorphisms of a compact, node-colored *problem description graph (PDG)*. As it was first observed by Pochter, Zohar, and Rosenschein (2011), every automorphism of Π's PDG explicitly induces an automorphism of $\mathcal{T}_\Pi$, and the former can be searched for using off-the-shelf tools for discovery of automorphisms in explicit, colored graphs, such as BLISS (Junttila and Kaski 2007). In addition, this search can be easily restricted to PDG automorphisms to stabilizers of S_*, that is, $\Gamma^{\mathrm{pdg}}_{S_*}$. Finally, since finding the precise equivalence relation $\sim_\Gamma$ induced by the discovered subgroup $\Gamma \leq \Gamma^{\mathrm{pdg}}_{S_*} \leq \Gamma_{S_*} \leq Aut(\mathcal{T}_\Pi)$ is NP-hard (Luks 1993), it is approximated (with a loss of precision, but not of correctness) via an equivalence relation $\sim \leq \sim_\Gamma$, defined by a heuristic local search in S, with the generators of Γ defining state neighborhood, and state evaluation being based on a lexicographic ordering of S (Pochter, Zohar, and Rosenschein 2011). Note that $s \sim s'$ implies $s' = \sigma(s)$ for $\sigma \in \Gamma$, which is derived from the local search paths from s and s' to the (same) canonical state.

Satisfying Planning with Γ_{S_*}

Enhancing optimal A^* planning with DKS has been shown empirically effective, not only for reduction in expanded nodes, but also for increasing the overall coverage (Domshlak, Katz, and Shleyfman 2012). In principle, nothing prevents us from adopting DKS in satisficing planning; this is true whether the planning is based on *GBFS*, on *WA**, or on an iterative combination of the two as in LAMA. However, it is not clear whether the overhead of reasoning about symmetries pays off in satisficing planning, and if so, what the right way is to incorporate this reasoning into the search process. This question initiated our investigation, and in what follows, we discuss both our initial findings and some subsequent developments.

As a first step, we have implemented DKS within both *GBFS* and *WA** iterations of LAMA. The extension, described in Figure 1, is independent of the heuristic function, eagerness of the state evaluation, and both preferred operators and heuristic composition mechanisms. The only two differences from DKS in A^* are that (i) in *GBFS*, s' is not reopened in step 2, and (ii) to cover both lazy and eager heuristic evaluations, step 2 considers state s not when it is generated, but when it is about to be evaluated.

The potential value of DKS in satisficing search is twofold. First, similarly to the effect obtained in A^*, no two states from the same equivalence class will be expanded. Second, the quality of the plans discovered with DKS is expected to be at least as good as, and possibly better than, the quality of the plans discovered without DKS. In particular:

Proposition 1 *Let* $\sim \leq \sim_{\Gamma_{S_*}}$*, and let* h *be a heuristic for a planning task* Π *that is invariant under* $\sim$*, i.e.,* $h(s) = h(s')$ *holds for all* $s \sim s'$*. Then, assuming perfect tie-breaking, if* π *and* π' *are plans for* Π *found by WA** *with and without reasoning about* $\sim$*, respectively, then* $cost(\pi) \leq cost(\pi')$*. Moreover, for any value of the WA** *weight parameter, it is possible that* $cost(\pi) < cost(\pi')$*.*

The claim also holds for *GBFS* as the latter can be considered as *WA** for a sufficiently large weight. To see how DKS can actually improve the plans, consider a schematic example of a state space in Figure 2, where s_7 is the initial state, $S_* = \{s_5, s_{10}\}$, and the solid arcs depict the transitions. There are two plans: the longer plan to s_5, and the shorter one, to s_{10}. It is easy to see that, for $i \in [5]$, we have $s_i \sim s_{i+5}$. Assuming heuristic values as above the state nodes in Figure 2, *GBFS* with lazy evaluation may generate the longer plan. With DKS, however, *GBFS* in such a case will necessarily evaluate s_8 before evaluating s_4. State s_8 will then be found symmetric to the previously evaluated s_3, causing the initial state s_7 to "adopt" s_3 (dotted arc). At the end, when s_5 is reached, the plan extracted by `trace-forward` from the "plan to s_5" will actually be the shorter plan to s_{10}.

Table 1 compares the performance of LAMA's *GBFS* with and without the DKS extension on IPC benchmark tasks; the former is denoted in Table 1 by *GBFS*$(\sim)$. The experiments were performed on the Intel(R) Xeon(R) CPU X5690 machine. Each task/planner was given a total time limit of 30 minutes, memory limit of 2 GB, and the search for automorphisms of its PDG was restricted to three minutes. The overall comparison in Table 1 is made on all IPC tasks in which the set Σ of group generators was not found empty; the number of such tasks per domain appears by the name of the domain. For both planners, Table 1 summarizes their performance in terms of task coverage, plan length[1], and node expansions. For plan length and expanded node measures, only tasks solved by both planners are considered. Bold font indicates strictly superior performance in the respective domains. Though the table speaks for itself, its

domain	coverage		length		expansions	
	GBFS	*GBFS*$(\sim)$	*GBFS*	*GBFS*$(\sim)$	*GBFS*	*GBFS*$(\sim)$
airport (23)	23	23	2788	2788	138846	**132445**
barman-11 (20)	20	20	3749	**3371**	1044225	**110279**
depot (21)	21	**22**	1238	**1213**	1428539	**355757**
driverlog (20)	20	20	1289	**1261**	125044	**123074**
elevators-08 (30)	30	30	2844	**2832**	32987	**32800**
elevators-11 (20)	20	20	4633	**4618**	147577	**143431**
floortile-11 (5)	5	5	**234**	237	5532003	**2201147**
freecell (1)	1	1	9	9	9	9
grid (5)	5	5	339	**331**	**1049**	1065
gripper (20)	20	20	1360	1360	1610	1610
logistics-00 (28)	28	28	1209	**1208**	5940	**5907**
logistics-98 (35)	35	35	3827	**3819**	109184	**108210**
miconic (141)	141	141	9435	**9434**	49041	**44061**
mprime (35)	35	35	314	314	1517	**1512**
mystery (19)	19	19	161	161	648165	**280149**
nomystery-11 (13)	13	**14**	451	451	43952	**34158**
openstacks-08 (30)	30	30	4282	4282	4282	4282
openstacks-11 (15)	15	15	6063	6063	6063	6063
openstacks (11)	11	11	1348	1348	1385	1385
parking-11 (20)	20	20	**1494**	1501	27755	**23503**
pathways (23)	23	23	2881	2881	33602	**33601**
pegsol-08 (30)	30	30	767	**762**	3812975	**3318054**
pegsol-11 (20)	20	20	644	**642**	3812951	**3317951**
pipes-notank (44)	44	44	**2980**	3250	474042	**292981**
pipes-tank (41)	41	**43**	2055	**1996**	3654131	**1209040**
psr-small (27)	27	27	550	550	11695	**9342**
rovers (6)	6	6	176	176	611	**574**
satellite (35)	35	35	4363	**4098**	129503	**107290**
scanalyzer-08 (26)	26	26	925	**923**	8822	**3647**
scanalyzer-11 (19)	19	19	801	**799**	8258	**3432**
sokoban-08 (25)	25	**27**	6139	**5569**	9842674	**2912204**
sokoban-11 (16)	16	**18**	4699	**4157**	9635861	**2796591**
tpp (29)	29	29	3582	**3530**	37389	**32330**
transport-08 (29)	**30**	29	2901	**2824**	**97669**	111414
transport-11 (15)	16	16	3343	**3270**	**185336**	196007
visitall-11 (20)	20	20	28776	**28702**	132939	**132937**
woodwork-08 (20)	20	20	996	**988**	176549	**169648**
woodwork-11 (4)	4	4	316	**315**	94880	**93351**
zenotravel (19)	19	19	731	**723**	9072	**9007**
TOTAL	962	**969**	114692	**112756**	41508132	**18360248**

Table 1: Performance of *GBFS* with and without DKS in terms of coverage, plan length, and expanded nodes.

overall message is that, across all the three measures, adopting DKS almost consistently improved *GBFS* performance. On the tasks solved by both planners, the total number of expanded nodes was cut by more than half, and the quality of the plans was improved in 24 out of 37 domains (and negatively affected only in 3 domains), with the most profound improvement in plan length being in the Sokoban domain.

Landmark Heuristic and Γ_{S_*}

While the results show the pros of employing DKS in satisficing search, pruning symmetric states can also be detriment if the heuristics in use are not invariant under $\sim_{\Gamma_{S_*}}$, as is the case with both the FF and landmark heuristics used by LAMA. It is always possible that, on the states of some equivalence class, the heuristic is most inaccurate on the state that is evaluated first by the search procedure. Given that the rest of that equivalence class will be pruned, it is possible that the pruning takes the planning into a much longer search than what it would undergo without pruning. At a first view, a repair suggests itself almost immediately: in step 2, before discarding state s, compute $h(s)$ and use it to update $h(s')$. The difficulty with that repair is twofold. First, even if both $h(s)$ and $h(s')$ are computed, which of them should be used for s' is somewhat clear only if h is admissible, while most of the heuristics used to date in satisficing planning are inadmissible. Second, whether the heuristic evaluation is lazy or eager, computing heuristic values for pruned states eliminates part of the value that symmetry

[1]Similarly to the *GBFS*-based first iteration of LAMA, the cost of the actions here is ignored and they are all treated as unit-cost.

domain	Lazy A^* (expansions)				LAMA (IPC quality score)			
	orig	I	II	III	orig	I	II	III
airport	839535	786870	786870	786870	23.00	23.00	23.00	23.00
barman-11	NA	NA	NA	NA	18.75	19.81	19.81	19.81
depot	4070390	2335741	**2137793**	**2178454**	20.56	21.46	**21.77**	21.66
driverlog	697132	401819	**399984**	**399922**	20.00	19.97	20.00	20.00
elevators-08	2302384	1157581	**1149631**	**1152269**	**29.47**	28.90	28.90	28.90
elevators-11	NA	NA	NA	NA	19.77	19.85	19.85	19.85
floortile-11	2752081	1815829	1817070	1815879	4.58	4.97	4.97	4.97
freecell	80	53	53	53	1.00	1.00	1.00	1.00
grid	3139	3139	3139	3139	**5.00**	4.98	4.98	4.98
gripper	2438675	294	294	294	20.00	20.00	20.00	20.00
logistics00	5427655	3639363	**3639362**	3639363	**27.98**	27.92	27.92	27.92
logistics98	2669671	315109	**315106**	**315095**	34.85	34.94	**34.96**	34.91
miconic	8855004	4647164	4655993	4655993	140.08	140.69	140.71	**140.71**
mprime	8007	7595	7595	7595	35.00	35.00	35.00	35.00
mystery	541656	219509	219509	219509	19.00	19.00	19.00	19.00
nomystery-11	1600874	884165	884168	**879411**	12.97	**14.00**	13.00	13.00
openstacks-08	18839	17399	17399	17399	29.85	29.87	30.00	30.00
openstacks-11	76249	71155	71155	71155	14.86	14.81	14.92	**14.94**
openstacks	3394	888	888	888	11.00	11.00	11.00	11.00
parking-11	209682	172025	178416	172025	19.70	19.93	19.73	**19.97**
pathways	1212690	1094981	1094981	1094981	23.00	23.00	23.00	23.00
pegsol-08	399516	348756	**348592**	348758	**29.82**	29.25	29.25	29.25
pegsol-11	395551	344983	**344814**	344984	**19.82**	19.25	19.25	19.25
pipes-notank	6124355	2403721	**2400164**	**2402995**	43.49	43.75	43.75	43.75
pipes-tank	6837018	1108446	1112253	1113753	36.76	42.73	42.47	**43.73**
psr-small	321959	119794	119794	119794	27.00	27.00	27.00	27.00
rovers	59436	57544	57544	57544	6.00	6.00	6.00	6.00
satellite	418675	191257	277613	277613	34.28	34.96	35.00	35.00
scanalyzer-08	1558042	16618	16622	16618	24.97	**25.19**	25.16	25.11
scanalyzer-11	1546069	15547	15548	15547	17.94	17.78	17.86	**18.23**
sokoban-08	7597615	4412720	**4339384**	**4376488**	24.00	27.00	27.00	27.00
sokoban-11	7243596	4156083	**4102072**	**4119717**	15.00	17.98	18.00	18.00
tpp	97675	19914	19914	19914	**28.80**	28.67	28.67	28.67
transport-08	927243	278964	280472	280763	**29.82**	28.29	29.52	29.52
transport-11	NA	NA	NA	NA	**15.94**	15.60	15.60	14.55
visitall-11	NA	NA	NA	NA	19.99	19.99	19.99	19.99
woodwork-08	63652	31363	31363	31363	19.76	20.00	20.00	20.00
woodwork-11	NA	NA	NA	NA	3.99	4.00	4.00	4.00
zenotravel	733892	626391	626668	626485	18.74	19.00	19.00	19.00
TOTAL	68051431	31702780	**31472223**	**31562630**	946.54	960.53	961.06	**961.68**

Table 2: Lazy A^* and LAMA, with and without multi-state inference for the landmark heuristic.

breaking brings to the search process in the first place.

We now show that, at least with the landmark heuristic, heuristic-related information between symmetric states can be communicated in a meaningful and cost-effective way. The basic idea corresponds to extending the multi-path inference of landmarks to "multi-state" inference between the symmetric states. Recall that each $\sigma \in \Gamma^{\text{pdg}}_{S_*}$ maps variable assignments to variable assignments. Let L be a set of disjunctive landmarks for Π, and let $\Gamma \leq \Gamma^{\text{pdg}}_{S_*}$. For each landmark $\varphi \in L$, and each $\sigma \in \Gamma$, by $\sigma(\varphi)$ we denote the set of variable assignments obtained by applying σ to each of the variable assignments in φ, that is, $\sigma(\varphi) = \{\langle \sigma(v), \sigma(d) \rangle \mid \langle v, d \rangle \in \varphi\}$. Similarly, by $\sigma(L')$ for $L' \subseteq L$, we denote the set $\{\sigma(\varphi) \mid \varphi \in L\}$.

Proposition 2 *Let Π be a planning task, $s, s' \in S$, and $\Gamma \leq \Gamma^{pdg}_{S_*}$. If $s' = \sigma(s)$ for some $\sigma \in \Gamma$, and φ is a landmark for $\Pi(s)$, then $\sigma(\varphi)$ is a landmark for $\Pi(s')$.*

The proof of Proposition 2 is almost immediate from the definitions of $\Gamma^{\text{pdg}}_{S_*}$ and landmarks. Note also that Proposition 2 is independent of how landmarks for different states of Π are discovered in the first place. In particular, landmarks for $\Pi(s)$ can either be restricted, as in LAMA, to the landmarks for $\Pi \equiv \Pi(s_0)$, or discovered specifically for $\Pi(s)$ (Helmert and Domshlak 2009; Bonet and Helmert 2010). In LAMA extended with DKS as in Figure 1, at step 2 we can update the set of landmarks $L(s')$ to be achieved from s onwards to $L(s') \cup \sigma(L(s))$. That is, if the search starts with a set L of landmarks for Π, then the sets of yet to be achieved landmarks $L(s)$ for states s of Π are no longer restricted to subsets of L, but to subsets of a (possibly much larger) set $\{\sigma(\varphi) \mid \varphi \in L, \sigma \in \Gamma^{\text{pdg}}_{S_*}\} \supseteq L$.

First, landmarks of state s reached by LAMA are computed at a very low effort from the landmarks of its parent state and the respective operator. Hence, the computational overhead of the multi-state inference of landmarks between the symmetric states remains low. Second, updating the heuristic estimate of the equivalence class representative s' this way results in a more accurate estimate of s', *subject to* validity of the assumption that knowing more landmarks of $\Pi(s')$ results in more accurate estimates of the goal distance from s'. This assumption does not hold for the landmark heuristic in general (or otherwise the latter would be admissible), but it is still the core assumption behind the landmark heuristic, similarly to how the "shorter relaxed plans are more accurate" assumption underlies the FF heuristic. Hence, multi-state inference of landmarks between the symmetric states is at least fully consistent with the concept of LAMA's landmark heuristic.

While Proposition 2 allows for inferring landmarks for $\Pi(s)$ that are not (or are not known to be) landmarks of Π, in our current extension of LAMA we restrict our inference to the initially discovered landmarks L of Π. The resulting modification of step 2 in Figure 1 is summarized by the following corollary of Proposition 2:

Corollary 3 Let L be a set of landmarks for a planning task Π, Γ be a subgroup of $\Gamma^{\text{pdg}}_{S_*}$, and $\varphi \in L$ be a landmark of Π. For any pair of states s and s', if $s' = \sigma(s)$ for some $\sigma \in \Gamma$, $\varphi \in L(s)$, and $\sigma(\varphi) \in L$, then $\sigma(\varphi) \in L(s')$.

Denoting the basic DKS in Table 2 by (I), we denote by (II) DKS with the landmark heuristic enhanced with the above corollary. We have also implemented a more restricted, but also less costly, version of (II), denoted by (III), which relies only on landmarks in L that are fixpoints of the subgroup $\Gamma \leq \Gamma^{\text{pdg}}_{S_*}$ in use. Note that (II) and (III) bring value only for states that can possibly be expanded after their landmarks are updated by the inference. Thus, the impact of these approaches decreases with the search greediness, being the smallest for the *GBFS* iteration. Hence, first we have separately evaluated the marginal impact of II and III within A^* with lazy evaluation, which constitutes the last possible and the least greedy iteration of LAMA. The performance in terms of expanded nodes is shown in columns 2-5 of Table 2. Note that (I) always outperforms the baseline, and it is outperformed by both (II) and (III). Second, columns 6-9 show the IPC quality score performance of the baseline (LAMA), as well as its extensions to (I), (II), and (III), within the LAMA's iterative search process. Here as well, all three extensions are shown to be cost-effective.

Acknowledgments. The work of Domshlak and Shleyfman was partially supported by ISF grant 1045/12 and the Technion-Microsoft E-Commerce Research Center.

References

Bäckström, C., and Klein, I. 1991. Planning in polynomial time: The SAS-PUBS class. *Comp. Intell.* 7(3):181–197.

Bäckström, C., and Nebel, B. 1995. Complexity results for SAS$^+$ planning. *Comp. Intell.* 11(4):625–655.

Bonet, B., and Helmert, M. 2010. Strengthening landmark heuristics via hitting sets. In *ECAI*, 329–334.

Domshlak, C.; Katz, M.; and Shleyfman, A. 2012. Enhanced symmetry breaking in cost-optimal planning as forward search. In *ICAPS*, *to appear*.

Emerson, E. A., and Sistla, A. P. 2011. Symmetry and model checking. *Formal Methods in System Design* 9(1-2):105–131.

Fox, M., and Long, D. 1999. The detection and exploitation of symmetry in planning problems. In *IJCAI*, 956–961.

Fox, M., and Long, D. 2002. Extending the exploitation of symmetries in planning. In *AIPS*, 83–91.

Helmert, M., and Domshlak, C. 2009. Landmarks, critical paths and abstractions: What's the difference anyway? In *ICAPS*, 162–169.

Helmert, M. 2006. The Fast Downward planning system. *JAIR* 26:191–246.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *JAIR* 14:253–302.

Junttila, T., and Kaski, P. 2007. Engineering an efficient canonical labeling tool for large and sparse graphs. In *ALENEX*, 135–149.

Karpas, E., and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *IJCAI*, 1728–1733.

Luks, E. M. 1993. Permutation groups and polynomial-time computation. In *Groups and Computation, DIMACS Series in Disc. Math. and Th. Comp. Sci.*, volume 11. 139–175.

Pochter, N.; Zohar, A.; and Rosenschein, J. S. 2011. Exploiting problem symmetries in state-based planners. In *AAAI*.

Puget, J.-F. 1993. On the satisfiability of symmetrical constrained satisfaction problems. In *ISMIS*, 350–361.

Richter, S., and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *JAIR* 39:127–177.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *AAAI*, 975–982.

Richter, S.; Thayer, J. T.; and Ruml, W. 2010. The joy of forgetting: Faster anytime search via restarting. In *ICAPS*, 137–144.

Richter, S.; Westphal, M.; and Helmert, M. 2011. LAMA 2008 and 2011 (planner abstract). In *IPC 2011, Deterministic Part*. 50–54.

Rintanen, J. 2003. Symmetry reduction for SAT representations of transition systems. In *ICAPS*, 32–41.