







exact. We now establish that in the new theory, label reduction is always safe.

Consider a label reduction with mapping  $\tau$  that transforms  $X = \{\Theta_1, \dots, \Theta_n\}$  into  $X' = \{\tau(\Theta_1), \dots, \tau(\Theta_n)\}$ . We first show that this label reduction is transition-safe. Here and in the following, we write states of  $\otimes X$  and  $\otimes X'$  as tuples  $\langle s_1, \dots, s_n \rangle$  where each  $s_i$  is a state of  $\Theta_i$ . Consider some transition  $\langle s_1, \dots, s_n \rangle \xrightarrow{l} \langle t_1, \dots, t_n \rangle \in \otimes X$ . By the definition of products, we have  $s_i \xrightarrow{l} t_i \in \Theta_i$  for all  $1 \leq i \leq n$ ; by the definition of label reduction, we have  $s_i \xrightarrow{\tau(l)} t_i \in \tau(\Theta_i)$  for all  $1 \leq i \leq n$ ; finally, again by definition of products we have  $\langle s_1, \dots, s_n \rangle \xrightarrow{\tau(l)} \langle t_1, \dots, t_n \rangle \in \otimes X'$ . With  $\alpha$  set to the identity function, this proves that label reduction is transition-safe. (Label reduction does not change the set of goal states.) Due to the condition on  $\tau$  in the definition of label reduction, the transformation is also cost-safe. In summary, label reduction is safe.

### Exact Label Reduction

Previous papers that study label reduction in the merge-and-shrink framework (Nissim, Hoffmann, and Helmert 2011a; Helmert et al. 2014) focus on the question which conditions are required to make label reduction exact. In particular, exact label reduction is a critical ingredient in the polynomial-time perfect heuristics obtained in some planning domains (Nissim, Hoffmann, and Helmert 2011a).

Helmert et al. (2014) discuss conditions for exactness of label reduction that are sufficient and in a certain sense necessary, thus seemingly closing the topic of exact label reduction. However, these results do not directly apply to our theory, as they rely on the limitations of the previous theory. We revisit the topic here, proving a sufficient and necessary condition for exact label reduction that generalizes the previous result.

It is obvious that a label reduction is cost-exact iff it only combines labels of the same cost (i. e.,  $\tau(l) = \tau(l')$  implies  $c(l) = c(l')$ ), and of course we must always set  $c(\tau(l)) := c(l)$  to be cost-exact. It remains to discuss under which conditions label reduction is transition-exact. We start by introducing some additional terminology.

**Definition 1.** Let  $X$  be a set of transition systems with labels  $L$ . Let  $l, l' \in L$  be labels, and let  $\Theta \in X$ .

Label  $l$  is alive in  $X$  if all transition systems  $\Theta' \in X$  have some transition  $s \xrightarrow{l} t \in \Theta'$ . Otherwise,  $l$  is dead.

Label  $l$  locally subsumes label  $l'$  in  $\Theta$  if for all  $s \xrightarrow{l'} t \in \Theta$  we also have  $s \xrightarrow{l} t$ . Label  $l$  globally subsumes label  $l'$  in  $X$  if  $l$  locally subsumes  $l'$  in all  $\Theta' \in X$ .

Labels  $l$  and  $l'$  are locally equivalent in  $\Theta$  if they label the same transitions in  $\Theta$ , i. e., if  $l$  and  $l'$  locally subsume each other in  $\Theta$ .

Labels  $l$  and  $l'$  are  $\Theta$ -combinable in  $X$  if they are locally equivalent in all transition systems  $\Theta' \in X \setminus \{\Theta\}$ . (It does not matter whether or not they are locally equivalent in  $\Theta$ .)

It is easy to see that dead labels induce no transitions in  $\otimes X$ . Consequently, it is an exact transformation to remove all dead labels (and their transitions) from  $X$ . Hence, it suffices to consider the case where  $X$  has no dead labels.

Moreover, we can restrict attention to label reductions  $\tau$  that combine two labels  $l_1$  and  $l_2$  into some new label  $l_{12}$  ( $\tau(l_1) = \tau(l_2) = l_{12}$ ) while leaving all other labels unchanged ( $\tau(l') = l'$  for all  $l' \notin \{l_1, l_2\}$ ). Other label reductions can be represented as chains of such “minimal” label reductions. We are now ready to state our major result.

**Theorem 1.** Let  $X$  be a set of transition systems without dead labels. Consider a label reduction on  $X$  which combines labels  $l_1$  and  $l_2$  and leaves other labels unchanged.

This label reduction is exact iff  $c(l_1) = c(l_2)$  and

1.  $l_1$  globally subsumes  $l_2$ , or
2.  $l_2$  globally subsumes  $l_1$ , or
3.  $l_1$  and  $l_2$  are  $\Theta$ -combinable for some  $\Theta \in X$ .

*Proof.* Let  $\tau$  be the described label mapping, let  $X = \{\Theta_1, \dots, \Theta_n\}$  and let  $X' = \{\tau(\Theta_1), \dots, \tau(\Theta_n)\}$  be the result of label reduction. Let  $l_{12} := \tau(l_1) = \tau(l_2)$ .

Clearly, the label reduction is cost-exact iff  $c(l_1) = c(l_2)$ . We need to show that it is transition-exact iff 1., 2., or 3. holds. We prove this in three parts:

- (A) If neither 1. nor 2. nor 3. holds, then the label reduction is not exact.
- (B) If 1. or 2. holds, then the label reduction is exact.
- (C) If 3. holds, then the label reduction is exact.

Label reduction is always transition-safe and leaves the set of goal states unchanged, so we only need to consider the second condition in the definition of transition-exactness.

On (A): We must show that no function  $\alpha$  satisfies the criterion of transition-exactness. It is sufficient to consider the case where  $\alpha$  is a bijection because  $\otimes X$  and  $\otimes X'$  have the same number of states, so non-bijective  $\alpha$  cannot possibly work. Renaming states does not affect the notion of exactness, so we can further limit attention to  $\alpha$  being the identity function without loss of generality.

We say that a transition system  $\Theta \in X$  has an  $l_1$ -only transition if there exists a transition  $s \xrightarrow{l_1} t \in \Theta$  with  $s \xrightarrow{l_2} t \notin \Theta$ . Symmetrically, it has an  $l_2$ -only transition if there exists a transition  $s \xrightarrow{l_2} t \in \Theta$  with  $s \xrightarrow{l_1} t \notin \Theta$ .

We try to find two transition systems  $\Theta_i, \Theta_j \in X$  with  $i \neq j$  such that there is an  $l_1$ -only transition  $s_i \xrightarrow{l_1} t_i \in \Theta_i$  and an  $l_2$ -only transition  $s_j \xrightarrow{l_2} t_j \in \Theta_j$ . Then  $\Theta_i \otimes \Theta_j$  does not contain a transition  $\langle s_i, s_j \rangle \xrightarrow{l} \langle t_i, t_j \rangle$  for either  $l = l_1$  or  $l = l_2$ , but  $\tau(\Theta_i) \otimes \tau(\Theta_j)$  does contain the transition  $\langle s_i, s_j \rangle \xrightarrow{l_{12}} \langle t_i, t_j \rangle$ . By induction over the remaining transition systems, it is then easy to show that  $\otimes X'$  contains a transition that does not correspond to a transition in  $\otimes X$ , proving inexactness. (Here, we use that there are no dead labels: the argument fails if  $l_1$  and  $l_2$  are dead.) It remains to show that  $l_1$ -only and  $l_2$ -only transitions in different transition systems of  $X$  exist.

Because 1. does not hold, there exists an  $l_2$ -only transition in some transition system  $\Theta \in X$ . Because 2. does not hold, there exists an  $l_1$ -only transition in some transition system  $\Theta' \in X$ . If  $\Theta$  and  $\Theta'$  are different transition systems, we have found the required transitions and are done.

So let us assume that  $\Theta = \Theta'$ . Because 3. does not hold, there exist at least two transition systems where  $l_1$  and  $l_2$

are not locally equivalent, so there is at least one transition system  $\Theta'' \neq \Theta$  where they are not locally equivalent. This means that  $\Theta''$  must have an  $l_1$ -only transition or an  $l_2$ -only transition. In the former case, we select the  $l_1$ -only transition in  $\Theta''$  and the  $l_2$ -only transition in  $\Theta$ . Otherwise, we select the  $l_2$ -only transition in  $\Theta''$  and the  $l_1$ -only transition in  $\Theta'$  ( $= \Theta$ ).

On (B): Consider Case 1., where  $l_1$  globally subsumes  $l_2$ . Case 2. is identical with  $l_1$  and  $l_2$  swapped. As the function  $\alpha$  in the definition of transition-exactness, we choose the identity mapping. Then the condition for transition-exactness we need to verify simplifies to: for all  $s \xrightarrow{l'} t \in \otimes X'$ , there exists a label  $l \in \tau^{-1}(l')$  with  $s \xrightarrow{l} t \in \otimes X$ . For  $l' \neq l_{12}$ , this is trivial because  $\otimes X$  and  $\otimes X'$  are exactly identical regarding labels other than  $l_1, l_2$  and  $l_{12}$ . So consider the case  $l' = l_{12}$ . Let  $s = \langle s_1, \dots, s_n \rangle$  and let  $t = \langle t_1, \dots, t_n \rangle$ . From  $s \xrightarrow{l_{12}} t \in \otimes X'$  we get  $s_i \xrightarrow{l_{12}} t_i \in \tau(\Theta_i)$  for all  $1 \leq i \leq n$ , and hence  $s_i \xrightarrow{l_1} t_i \in \Theta_i$  or  $s_i \xrightarrow{l_2} t_i \in \Theta_i$  for all  $1 \leq i \leq n$ . Because  $l_1$  globally subsumes  $l_2$ , this implies  $s_i \xrightarrow{l_1} t_i \in \Theta_i$  for all  $1 \leq i \leq n$ , and hence  $s \xrightarrow{l_1} t \in \otimes X$ , concluding this part of the proof.

On (C): As in (B), we set  $\alpha$  to the identity function and only need to consider transitions  $s \xrightarrow{l_{12}} t \in \otimes X'$ . Let  $s = \langle s_1, \dots, s_n \rangle$  and let  $t = \langle t_1, \dots, t_n \rangle$ . Again, we obtain that for all  $1 \leq i \leq n$ ,  $s_i \xrightarrow{l_{12}} t_i$  and hence  $s_i \xrightarrow{l_1} t_i$  or  $s_i \xrightarrow{l_2} t_i$ . Choose  $l \in \{l_1, l_2\}$  such that  $s_j \xrightarrow{l} t_j$ , where  $j \in \{1, \dots, n\}$  is chosen in such a way that  $l_1$  and  $l_2$  are  $\Theta_j$ -combinable in  $X$ . (Such a transition system  $\Theta_j$  exists because we are in Case 3.) By the definition of  $\Theta$ -combinable,  $l_1$  and  $l_2$  are locally equivalent for all transition systems in  $X$  other than  $\Theta_j$ , and hence  $(s_i \xrightarrow{l_1} t_i$  or  $s_i \xrightarrow{l_2} t_i)$  implies  $(s_i \xrightarrow{l_1} t_i$  and  $s_i \xrightarrow{l_2} t_i)$  for all  $i \neq j$ . This shows that  $s_i \xrightarrow{l} t_i \in \Theta_i$  for all  $1 \leq i \leq n$ , and hence  $s \xrightarrow{l} t \in \otimes X$ , concluding the final part of the proof.  $\square$

We conclude the section with a brief discussion of the conditions in Theorem 1. Although all conditions can be checked in low-order polynomial time, there is a practical difference in complexity. Finding  $\Theta$ -combinable labels essentially consists in computing the local equivalence relations of all  $\Theta \in X$ , which is possible in linear time in the representation size of  $X$ . In contrast, finding globally subsumed labels involves finding subset relationships in a set family, for which to the best of our knowledge no linear-time algorithms are known.

A comparison to the results of Helmert et al. (2014) shows that the  $\Theta$ -combinability condition strictly generalizes the previous conditions on exactness. Hence, the new theory permits a larger number of exact label reductions even if we only use  $\Theta$ -combinability and do not consider global subsumption of labels. For this reason, coupled with efficiency concerns, we only perform exact label reductions based on  $\Theta$ -combinability in our experiments, which we describe next.

## Experiments

As discussed in the preceding sections, the new theory of label reduction is significantly more general and at the same time much less complicated than previous work. However,

we have yet to establish that it is useful for practical implementations of merge-and-shrink heuristics.

Firstly, we need to show that label reduction is actually a practically useful element of the merge-and-shrink toolbox. Although previous papers on merge-and-shrink heuristics already mentioned significant performance improvements due to label reduction, these are not a central focus of any previous experiment, and we think it is important to give solid quantitative evidence in favour of label reduction.

Secondly, while the semantic (rather than syntax-based, as in previous work) basis for exact label reduction has the advantage of being much more flexible and easier to implement than previous label reduction theory, it does carry a nontrivial computational overhead. If this overhead were so large that implementations based on the new theory performed significantly worse than ones based on the older theory, the usefulness of the new theory would be diminished.

Thirdly, a major drawback of previous label-reduction approaches are the limitations and difficulties in using them for *non-linear* merge strategies. Consequently, we are not aware of any implementations of non-linear merge strategies in the planning literature. The new theory removes these weaknesses, so it is appropriate to test it with non-linear merging.

In this section, we report on experiments that address these three aspects.

## Experiment Description

Our experiments were conducted with the Fast Downward planning system (Helmert 2006), which already features the merge-and-shrink framework including the previous label reduction approach. We evaluate on all benchmarks from the International Planning Competitions for optimal planning (up to 2011) that only use language features supported by the merge-and-shrink framework (44 domains and 1396 instances in total). The experiments were performed on Intel Xeon E5-2660 CPUs running at 2.2 GHz, using a time bound of 30 minutes and a memory bound of 2 GB per run.

All planning algorithms we evaluate employ an  $A^*$  search with a merge-and-shrink heuristic, which we varied along three dimensions: *label reduction method*, *merge strategy* and *shrink strategy*.

**Label Reduction Methods** We consider the case without label reduction (*none*), the *old* label reduction method based on the syntactic descriptions of operators (Nissim, Hoffmann, and Helmert 2011a; Helmert et al. 2014) and the *new* concept of label reduction described in this paper.

Our implementation of the new method only performs exact label reduction, combining labels whenever the  $\Theta$ -combinability condition in Theorem 1 applies. Specifically, the computation proceeds as follows: whenever label reduction makes sense (after each merge or shrink step), we compute the local equivalence relations for labels in each transition system, then use these to test for  $\Theta$ -combinable labels in each transition system  $\Theta$ . If such labels exist, they are combined in all transition systems, and the local equivalence relations are recomputed. The process repeats until no further  $\Theta$ -combinable labels exist for any transition system  $\Theta$ . Local equivalence relations are cached so that they are only

recomputed from scratch if the given transition system has changed since the last computation.

**Merge Strategies** We consider two merge strategies. Firstly, in order to represent the state of the art, we report results for the (linear) *reverse-level* (RL) strategy used in previous work (Nissim, Hoffmann, and Helmert 2011a; 2011b).

However, to more fully utilize the potential of the new label reduction approach, we also evaluate it on a *non-linear* merge strategy, for which the previous label reduction approach is comparatively ill-suited and no implementations were previously available. Therefore, as a case study, we implemented the originally proposed non-linear strategy by Dräger, Finkbeiner, and Podelski (2006) from model checking, which we call the *DFP* merge strategy in the following.

Roughly speaking, the DFP merge strategy is based on the idea of preferably merging transition systems which must synchronize on labels that occur close to a goal state. We refer to the original paper by Dräger et al. (2006) for details. We remind the reader that the work of Dräger et al. preceded the concept of label reduction, so the combination of non-linear merge strategies with label reduction is novel.

**Shrink Strategies** We report results on shrink strategies based on bisimulation (Nissim, Hoffmann, and Helmert 2011a; Helmert et al. 2014), which set the current state of the art. Specifically, we consider a shrink strategy based on *greedy bisimulation* with no limit on transition system size ( $G-N\infty$ ) as well as shrink strategies based on (exact) *bisimulation* with different size limits  $N$  for the intermediate transition system size ( $B-N10k$ ,  $B-N50k$ ,  $B-N100k$ ,  $B-N200k$ ,  $B-N\infty$ ). For example, with  $N = 10000$  (strategy  $B-N10k$ ), shrinking is performed to guarantee that no intermediate transition system has more than 10,000 abstract states, while with  $N = \infty$  (strategy  $B-N\infty$ ) there is no size bound, so that a perfect heuristic is constructed.

The *threshold* parameter (Helmert et al. 2014) was set to  $N$  for the strategies with bounded transition system size and to 1 for the unbounded ones ( $G-N\infty$  and  $B-N\infty$ ), following Nissim, Hoffmann, and Helmert (2011a). This configuration includes the shrink strategies used in the *merge-and-shrink* planner that participated in IPC 2011 (Nissim, Hoffmann, and Helmert 2011b).

## Experimental Results

Table 1 provides a result overview for coverage, i. e., the number of instances solved by each planner configuration within our resource bounds. The top half of the table presents results for the linear merge strategy (RL), the bottom half presents results for the non-linear DFP strategy.

**Usefulness of Label Reduction** Table 1 shows that planner configurations with label reduction dramatically outperform the corresponding ones without. (For readers less familiar with optimal planning, we point out that these tasks tend to scale exponentially in difficulty, so that even small improvements in coverage tend to be very hard to obtain.)

Table 2 shows detailed coverage results for the individual planning domains in the benchmark set for the best-

merge/shrink strategy	none	old	new
RL-G- $N\infty$	417	485	465
RL-B-N10k	590	624	617
RL-B-N50k	577	618	634
RL-B-N100k	560	599	<b>639</b>
RL-B-N200k	544	590	630
RL-B- $N\infty$	257	302	302
DFP-G- $N\infty$	415	—	465
DFP-B-N10k	597	—	622
DFP-B-N50k	565	—	<b>644</b>
DFP-B-N100k	551	—	632
DFP-B-N200k	522	—	625
DFP-B- $N\infty$	253	—	302

Table 1: Total coverage for several merge-and-shrink configurations, using no label reduction (*none*), the previous (*old*) or the *new* label reduction. See the text for descriptions of the merge and shrink strategies. Best results for each merge strategy in bold.

	RL-B-100K			DFP-B-50K	
	none	old	new	none	new
mprime (35)	8	+6	<b>+15</b>	6	<b>+17</b>
micronic (150)	60	<b>+13</b>	<b>+13</b>	58	<b>+14</b>
gripper (20)	7	<b>+13</b>	<b>+13</b>	7	<b>+11</b>
freecell (80)	6	-2	<b>+13</b>	9	<b>+11</b>
mystery (30)	8	+1	<b>+8</b>	8	<b>+8</b>
zenotravel (20)	9	<b>+3</b>	<b>+3</b>	10	<b>+2</b>
pipesworld-tankage (50)	8	+2	<b>+3</b>	12	<b>+2</b>
nomystery-opt11-strips (20)	17	+1	+1	16	+2
woodworking-opt08-strips (30)	11	-1	+1	11	+2
blocks (35)	<b>25</b>	-3	-3	25	<b>+2</b>
grid (5)	1	<b>+2</b>	<b>+2</b>	1	+1
floortile-opt11-strips (20)	5	+1	+1	4	+1
rovers (40)	7	+1	+1	7	+1
satellite (36)	5	+1	+1	5	+1
scanalyzer-08-strips (30)	12	+1	+1	12	+1
scanalyzer-opt11-strips (20)	9	+1	+1	9	+1
woodworking-opt11-strips (20)	6	-1	+1	6	+1
pipesworld-notankage (50)	<b>14</b>	$\pm 0$	$\pm 0$	14	+1
sokoban-opt08-strips (30)	24	$\pm 0$	<b>+2</b>	<b>25</b>	$\pm 0$
trucks-strips (30)	6	$\pm 0$	<b>+2</b>	6	$\pm 0$
transport-opt11-strips (20)	6	+1	+1	6	$\pm 0$
driverlog (20)	<b>13</b>	-1	-1	<b>12</b>	$\pm 0$
<b>Sum (791)</b>	267	+39	+79	269	+79
<b>Remaining domains (605)</b>	<b>293</b>	$\pm 0$	$\pm 0$	<b>296</b>	$\pm 0$
<b>Sum (1396)</b>	560	599	<b>639</b>	565	<b>644</b>

Table 2: Per-domain coverage. Columns 2–4 compare no (*none*), *old* and *new* label reduction for the linear merge strategy *reverse level* (RL) in its best configuration (RL-B-100K). Columns 5–6 compare no (*none*) and *new* label reduction for the non-linear DFP merge strategy in its best configuration (DFP-B-50K). For *old* and *new*, the columns show increase/decrease in coverage compared to *none*. Domains where label reduction showed no increase/decrease in coverage are omitted. The best results for the given merge strategy are highlighted in bold.

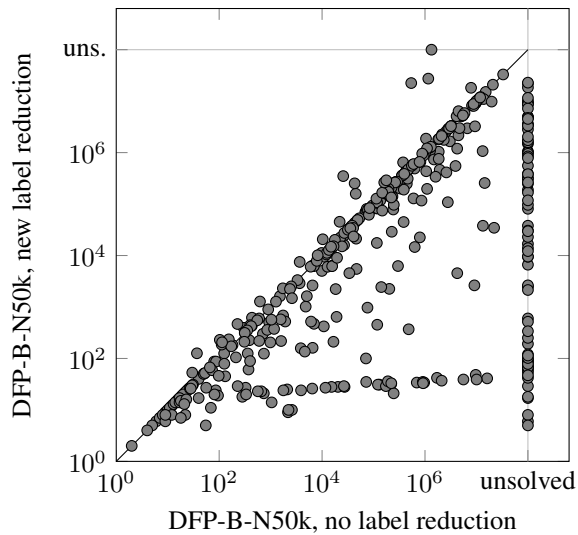


Figure 2: Number of expanded states for DFP-B-N50k: no label reduction vs. new label reduction.

performing shrink strategies for each merge strategy. The table shows that label reduction is very useful across the board, over a wide range of domains.

For the linear RL merge strategy, the new label reduction approach increases coverage in 19 domains compared to the baseline where no labels are reduced, while decreasing coverage in 2 domains. For the non-linear DFP merge strategy, label reduction increases coverage in 18 domains and decreases it in none.

To provide another detailed view, Figure 2 shows the number of expanded states with the strongest configuration, DFP-B-N50k, with and without label reduction. The figure plots the results without label reduction against the results with our new label reduction approach, over all instances in the benchmark suite. The figure clearly shows the significant impact that label reduction has on performance in many cases.

**Old vs. New Label Reduction Method** Focusing on the comparison between the old and new label reduction method with a linear merge strategy (top half of Table 1), we see that despite the larger effort involved in determining reducible labels, the results are in fact quite a bit *better* with new label reduction compared to the old technique. In particular, the best overall result of 639 solved tasks (RL-B-100k) is considerably higher than the best result with the previous state of the art (624 solved tasks with RL-B-10K and the old label reduction method).

There are two shrink strategies that show the opposite trend, namely the ones that tend to compute the simplest abstractions among the six strategies we consider: greedy bisimulation (RL-G-N $\infty$ ) and exact bisimulation with the smallest size bound (RL-B-N10k). One possible explanation for this behaviour is that for the shrink strategies that compute more complex abstractions, the additional label reductions afforded by the new method are critical for comput-

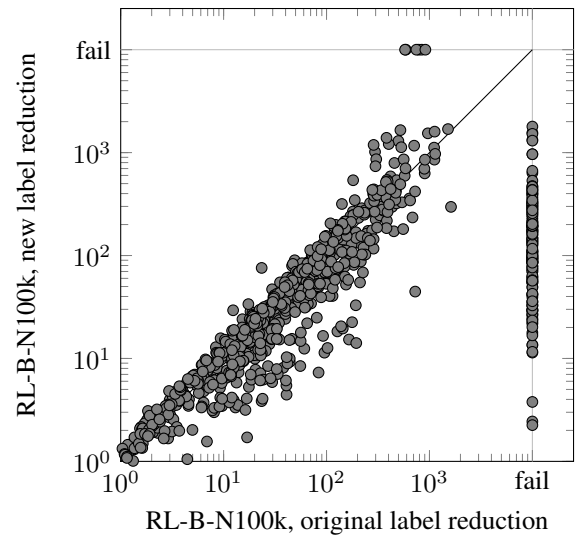


Figure 3: Construction time (in seconds) for RL-B-N100k: old label reduction vs. new label reduction. Almost all failures are due to running out of memory.

ing the merge-and-shrink abstraction within the given limits for time and especially memory. With the shrink strategies that compute simpler abstractions, on the other hand, memory for computing the abstraction is less of a concern, and the new label reduction method suffers from the higher computational cost for determining combinable labels.

This interpretation is supported by Figure 3, which compares the time to construct the abstraction heuristic for the old and new label reduction method for the strategy RL-B-N100k. The new strategy tends to construct abstractions faster and runs out of memory far less frequently. Figure 4 compares state expansions for the same configurations, showing that the heuristics are similarly informative in both cases, and it is mainly the ability to complete the computation of the abstraction (see Figure 3) that makes the difference between the old and new label reduction here.

In the case of perfect bisimulations (RL-B-N $\infty$ ), there is no difference in coverage between the two label reduction methods for a different reason: unless the given planning task exhibits significant amounts of symmetry, unrestricted bisimulation tends to exhaust the available memory very quickly, and hence the perfect abstraction heuristic is either computed quickly or not at all. In all cases not solved by the perfect bisimulation approaches, this is due to running out of memory while computing the abstraction.

**Non-Linear Merge Strategy** Shifting attention to the results for the non-linear DFP merge strategy (bottom half of Table 1), we see that the results with the new label reduction method are excellent. In particular, the best configuration (DFP-B-N50k) solves 644 tasks, again setting a new best result (compared to 639 solved by RL-B-N100k, also with our new label reduction method).

Generally speaking, the non-linear merge strategy appears to benefit even more from label reduction than the linear one

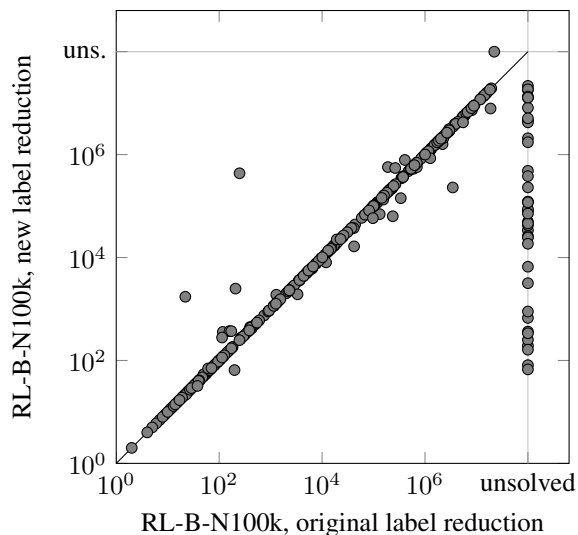


Figure 4: Number of expanded states for RL-B-N100k: old label reduction vs. new label reduction.

on average. One possible explanation for this observation is that non-linear merge strategies involve more complex products (merges) than linear ones, and hence benefit more from label reduction collapsing multiple parallel transitions into one. In linear merge strategies, at least one of the merged transition systems is always atomic, and atomic transition systems tend to have a comparatively low density of transitions. An alternative possibility is that label reduction interacts favourably with the DFP merge strategy, which – unlike merge strategies previously considered in planning – takes the labels into account directly in order to decide which transition systems to merge next.

Figure 5 compares the number of state expansions for the linear and non-linear merge strategy on an otherwise identical configuration (shrink strategy B-N50k, new label reduction). The comparison shows that the two merge strategies are quite complementary, with both strategies greatly outperforming each other on a significant number of instances.

## Conclusions

We have introduced a general theory of label reduction that addresses several drawbacks in the previous development of this topic. Compared to the previous theory, the new theory of label reduction is easier to understand, easier to reason about, and more general.

Under the new theory, label reduction can always be safely applied. Moreover, we have provided efficiently checkable necessary and sufficient criteria for label reduction to be *exact*, i. e., preserve all relevant information. The new theory allows identifying more cases where exact label reduction is possible, leading to improved performance of merge-and-shrink heuristics based on label reduction.

Unlike the previous theory of label reduction, the new theory allows for a straight-forward application of non-linear merge strategies. We conducted the first experiments of this

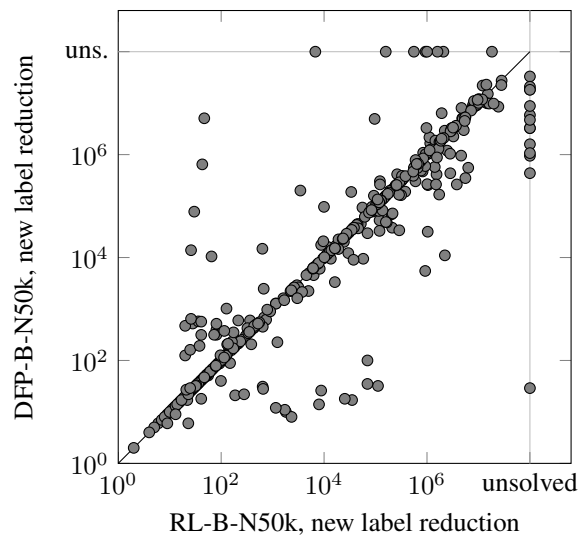


Figure 5: Number of expanded states for RL-B-N50k vs. DFP-B-N50k, both using new label reduction.

kind by adapting the originally proposed non-linear merge strategy from model checking to planning. In the future, we hope that the development of strong non-linear merge strategies can further increase the scalability of merge-and-shrink heuristics.

Another possible direction for future work is the exploration of *inexact* label reduction. Inexact label reduction is a general abstraction method just like shrinking, and similar intuitions to those that have guided the development of state-of-the-art shrink strategies could be used to develop useful inexact label reduction methods. For example, one might try to abstract a factored transition system by combining labels that only occur far away from goal states, similarly to the way that current shrink strategies prefer to combine abstract states that are far away from the goal.

## Acknowledgments

We thank the anonymous reviewers for their comments, which helped improve the paper. This work was supported by the Swiss National Science Foundation (SNSF) as part of the project “Abstraction Heuristics for Planning and Combinatorial Search” (AHPACS).

## References

- Bäckström, C., and Nebel, B. 1995. Complexity results for SAS<sup>+</sup> planning. *Computational Intelligence* 11(4):625–655.
- 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. *In-*



*ternational Journal on Software Tools for Technology Transfer* 11(1):27–37.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge-and-shrink abstraction: A method for generating lower bounds in factored state spaces. *Journal of the ACM*. In press.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2007. Flexible abstraction heuristics for optimal sequential planning. In Boddy, M.; Fox, M.; and Thiébaux, S., eds., *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 176–183. AAAI Press.

Helmert, M.; Haslum, P.; and Hoffmann, J. 2008. Explicit-state abstraction: A new method for generating heuristic functions. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, 1547–1550. AAAI Press.

Helmert, M. 2006. The Fast Downward planning system. *Journal of Artificial Intelligence Research* 26:191–246.

Nissim, R.; Hoffmann, J.; and Helmert, M. 2011a. 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 2011)*, 1983–1990.

Nissim, R.; Hoffmann, J.; and Helmert, M. 2011b. The Merge-and-Shrink planner: Bisimulation-based abstraction for optimal planning. In *IPC 2011 planner abstracts*, 106–107.