Cost Partitioning for Multi-agent Planning

Michal Štolba, Michaela Urbanovská, Daniel Fišer and Antonín Komenda
Department of Computer Science, Czech Technical University in Prague,
Karlovo náměstí 13, 121 35, Prague, Czech Republic

Keywords: Multi-agent Planning, Distributed Search, Distributed Heuristic Computation.

Abstract: Similarly to classical planning, heuristics play a crucial role in Multi-Agent Planning (MAP). Especially, the question of how to compute a distributed heuristic so that the information is shared effectively has been studied widely. This question becomes even more intriguing if we aim to preserve some degree of privacy, or admissibility of the heuristic. The works published so far aimed mostly at providing an ad-hoc distribution protocol for a particular heuristic. In this work, we propose a general framework for distributing heuristic computation based on the technique of cost partitioning. This allows the agents to compute their heuristic values separately and the global heuristic value as an admissible sum. We evaluate the presented techniques in comparison to the baseline of locally computed heuristics and show that the approach based on cost partitioning improves the heuristic quality over the baseline.

1 INTRODUCTION

Modern real-world large-scale personal, corporate or military applications often consist of multiple independent entities. Such entities may need to cooperate in the plan synthesis, while still wanting to protect the privacy of their input data and internal processes. Multi-agent and privacy-preserving multi-agent planning allow the definition of factors of the global planning problem private to the respective entities (i.e., agents) in order to improve the efficiency of planning and/or to maintain the privacy of the information.

In such privacy-preserving planning systems (Torreño et al., 2014; Nissim and Brafman, 2014; Maliah et al., 2016; Tožička et al., 2014), each agent has access only to its part of the global problem, thus can plan only using its operators. The agent can compute a heuristic estimate from its view of the global problem, i.e., its projection. Such projection also contains a view of other agents' public operators, which allows for a heuristic estimate of the entire problem, but such estimate may be significantly misguided as shown in (Štolba et al., 2015). The reason is that the projection does not take into account the parts of the problem private to other agents. Moreover in some problems, the optimal heuristic estimate may be arbitrarily lower for the projection than for the global problem.

To obtain a better guidance, a global heuristic estimate can be computed using a distributed process while, in some cases, still preserving privacy. The admissible distributed LM-Cut heuristic was proposed in (Štolba et al., 2015) and in (Maliah et al., 2015), the authors present a distributed admissible pattern database heuristic. Recently, a distributed variant of the class of potential heuristics has been proposed in (Štolba et al., 2016a).

MAD-A* (Nissim and Brafman, 2012) and its secure variant secure-MAFS (Brafman, 2015) are the only optimal privacy-preserving multi-agent planners. There is a number of optimal multi-agent planners not concerning privacy (Dimopoulos et al., 2012; Jezequel and Fabre, 2012).

All distributed heuristics published up-to-date present ad-hoc techniques to distribute each particular heuristic. Typically, the distributed computation of heuristic estimate requires the cooperation of all (or at least most of) the agents and incurs a substantial amount of communication. In many scenarios, the communication may be very costly (multi-robot systems) or prohibited (military) and even on high-speed networks, communication takes significant time compared to local computation. In such cases, it may pay off to use the projected heuristic instead of its better-informed counterpart. Most of the referenced heuristics are also missing any formal treatment of privacy, which is a nontrivial undertaking for such complex algorithms indeed.

Štolba, M., Urbanovská, M., Fišer, D. and Komenda, A.
Cost Partitioning for Multi-agent Planning.
DOI: 10.5220/0007256600400049
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 40-49
ISBN: 978-989-758-350-6

In (Nissim and Brafman, 2014), the authors mention an idea of an additive heuristic such that projected estimates of two agents could be added together and still maintain admissibility. In this paper, we apply the results of research of additive heuristics, namely the approach of cost partitioning, to the case of distribution of heuristics for multi-agent planning. This way we obtain a fully general approach allowing us to compute any heuristic additively in a distributed way. Also, it allows us to combine different heuristics, which adheres to the idea of independent agents (that is, each agent can use the heuristic it sees most fit). Last but not least, the presented approach allows us to compute an admissible sum of admissible heuristics.

In classical planning, the cost partitioning is typically computed for each state evaluated during the planning process. In privacy-preserving MAP, such approach does not make much sense as we want to keep the computation as local as possible. Thus, the envisioned use of such cost partitioning is to compute it once at the beginning of the planning process, use the cost partitioned problems to evaluate heuristics locally and sum the local heuristics to obtain a global estimate.

2 FORMALISM

In this section, we present the formalism used throughout the paper. First of all, we define a general (that is single-agent) planning task in the form of Multi-Valued Planning Task (MPT) (Helmert, 2006). The MPT is a tuple

$$\Pi = \left\langle \mathcal{V}, \mathcal{O}, s_I, s_\star, \mathsf{cost} \right\rangle$$

where $\mathcal{V}$ is a finite set of finite-domain variables, $\mathcal{O}$ is a finite set of operators, s_I is the initial state, $s_\star$ is the goal condition and $\mathsf{cost} : \mathcal{O} \mapsto \mathbb{R}_0^+$ is a cost function. Each V in the finite set of variables $\mathcal{V}$ has a finite domain of values $\mathsf{dom}(V)$. A *fact* $\langle V, v\rangle$ is a pair of a variable V and one of the values v from its domain (i.e. an assignment). Let p be a partial variable assignment over some set of variables $\mathcal{V}$. We use $\mathsf{vars}(p) \subseteq \mathcal{V}$ to denote a subset of $\mathcal{V}$ on which p is defined and $p[V]$ to denote the value of V assigned by p. Alternatively, p can be seen as a set of facts $\{\langle V, p[V]\rangle \,|V \in \mathsf{vars}(p)\}$ corresponding to that partial variable assignment. A complete assignment over $\mathcal{V}$ is a *state* over $\mathcal{V}$. A (partial) assignment p is *consistent* with an assignment p' iff $p[V] = p'[V]$ for all $V \in \mathsf{vars}(p)$.

An *operator* o from the finite set $\mathcal{O}$ has a precondition $\mathsf{pre}(o)$ and effect $\mathsf{eff}(o)$ which are both partial variable assignments. An operator o is applicable in a state s if $\mathsf{pre}(o)$ is consistent with s. Application of operator o in a state s results in a state s' such that all variables in $\mathsf{eff}(o)$ are assigned to the values in $\mathsf{eff}(o)$ and all other variables retain the values from s, formally $s' = o \circ s$.

A solution to MPT Π is a sequence $\pi = (o_1, ..., o_k)$ of operators from $\mathcal{O}$ (a plan), such that o_1 is applicable in $s_I = s_0$, for each $1 \leq l \leq k$, o_l is applicable in s_{l-1} and $s_l = o_l \circ s_{l-1}$ and s_k is a goal state (i.e., $s_\star$ is consistent with s_k).

Similarly as MA-STRIPS (Brafman and Domshlak, 2008) is an extension of STRIPS (Fikes and Nilsson, 1971) towards privacy and multi-agent planning, MA-MPT is a multi-agent extension of the Multi-Valued Planning Task. For n agents, the MA-MPT problem $\mathcal{M} = \{\Pi^i\}_{i=1}^n$ consists of a set of n MPTs. Each MPT for an agent $\alpha_i \in \mathcal{A}$ is a tuple

$$\Pi^i = \left\langle \mathcal{V}^i = \mathcal{V}^{\mathsf{pub}} \cup \mathcal{V}^{\mathsf{priv}_i}, \mathcal{O}^i = \mathcal{O}^{\mathsf{pub}_i} \cup \mathcal{O}^{\mathsf{priv}_i}, s_I^{\triangleright i}, s_\star^{\triangleright i}, \mathsf{cost}^i \right\rangle$$

where $\mathcal{V}^{\mathsf{priv}_i}$ is a set of private variables, $\mathcal{V}^{\mathsf{pub}}$ is a set of public variables shared among all agents $\mathcal{V}^{\mathsf{pub}} \cap \mathcal{V}^{\mathsf{priv}_i} = \emptyset$ and for each $i \neq j$, $\mathcal{V}^{\mathsf{priv}_i} \cap \mathcal{V}^{\mathsf{priv}_j} = \emptyset$ and $\mathcal{O}^i \cap \mathcal{O}^j = \emptyset$.

All variables in $\mathcal{V}^{\mathsf{pub}}$ and all values in their respective domain are public, that is known to all agents. All variables in $\mathcal{V}^{\mathsf{priv}_i}$ and all values in their respective domains are private to agent α_i which is the only agent aware of such V and allowed to modify its value.

A *global state* is a state over $\mathcal{V}^{\mathsf{G}} = \bigcup_{i \in 1..n} \mathcal{V}^i$. A global state represents the true state of the world, but no agent may be able to observe it as a whole. Instead, each agent works with an *i-projected state* which is a state over $\mathcal{V}^i$ such that all variables in $\mathcal{V}^{\mathsf{G}} \cap \mathcal{V}^i$ are equal in both assignments (the assignments are consistent).

The set $\mathcal{O}^i$ of operators of agent α_i consists of private and public operators such that $\mathcal{O}^{\mathsf{pub}_i} \cap \mathcal{O}^{\mathsf{priv}_i} = \emptyset$. The precondition $\mathsf{pre}(o)$ and effect $\mathsf{eff}(o)$ of private operators $o \in \mathcal{O}^{\mathsf{priv}_i}$, are partial assignments over $\mathcal{V}^{\mathsf{priv}_i}$, whereas in the case of public operators $o \in \mathcal{O}^{\mathsf{pub}_i}$ the assignment is over $\mathcal{V}^i$ and either $\mathsf{pre}(o)$ or $\mathsf{eff}(o)$ assigns a value to at least one public variable from $\mathcal{V}^{\mathsf{pub}}$. Because $\mathcal{V}^{\mathsf{pub}}$ is shared, public operators can influence (or be influenced by) other agents. The function $\mathsf{cost}^i : \mathcal{O}^i \mapsto \mathbb{R}_0^+$ assigns a cost to each operator of agent α_i. The initial state s_I and the partial goal state $s_\star$ (partial variable assignment over $\mathcal{V}^{\mathsf{G}}$) are in each agent's problem represented only as i-projected (partial) states.

We define a *global problem* (MPT) as a union of the agent problems, that is

$$\Pi^{\mathsf{G}} = \left\langle \bigcup_{i \in 1..n} \mathcal{V}^i, \bigcup_{i \in 1..n} \mathcal{O}^i, s_I, s_\star, \mathsf{cost}^{\mathsf{G}} \right\rangle$$

where $\mathsf{cost}^{\mathsf{G}}$ is a union of the cost functions cost^i. The global problem is the actual problem the agents are solving.

An *i-projected problem* is a complete view of agent α_i on the global problem Π^{G}. The i-projected problem of agent α_i contains i-projections of all operators of all agents. Formally, an i-projection $o^{\triangleright i}$ of $o \in O^i$ is o. For a public operator $o' \in O^{\mathsf{pub}_j}$ of some agent α_j s.t. $j \neq i$, an i-projected operator $o'^{\triangleright i}$ is o' with precondition and effect restricted to the variables of $\mathcal{V}^i$, that is $\mathsf{pre}(o'^{\triangleright i})$ is a partial variable assignment over $\mathcal{V}^i$ consistent with $\mathsf{pre}(o')$ ($\mathsf{eff}(o')$ treated analogously). An i-projection of a private operator $o'' \in O^{\mathsf{priv}_j}$ s.t. $j \neq i$ is $o''^{\triangleright i} = \varepsilon$, that is a no-op action with $\mathsf{cost}^{\triangleright i}(o''^{\triangleright i}) = \mathsf{cost}^i(\varepsilon) = 0$. The cost of i-projection of $o'' \in O^{\mathsf{pub}_j}$ is preserved, formally $\mathsf{cost}^{\triangleright i}(o^{\triangleright i}) = \mathsf{cost}^j(o)$.

The set of i-projected operators is

$$O^{\triangleright i} = \{o^{\triangleright i} | o \in \bigcup_{j \in 1 \ldots n} O^j\}$$

and an i-project*ed problem* is

$$\Pi^{\triangleright i} = \left\langle \mathcal{V}^i, O^{\triangleright i}, s_I^{\triangleright i}, s_\star^{\triangleright i}, \mathsf{cost}^{\triangleright i} \right\rangle$$

The set of all i-projected problems is then $\mathcal{M}^{\triangleright} = \{\Pi^{\triangleright i}\}_{i=1}^n$.

2.1 Example

Here we present a small running example with two agents α_1 and α_2. The problems Π^1 and Π^2 of agents α_1 and α_2 are:

agent:		α_1	α_2
$\mathcal{V}^{\mathsf{pub}}$	$=$	$\{V_3 \in \{u, g\}\}$	
$\mathcal{V}^{\mathsf{priv}_i}$	$=$	$\{V_1 \in \{i_1, p_1\}\}$	$\{V_2 \in \{i_2, p_2\}\}$
O^{pub_i}	$=$	$\{b_1\}$	$\{b_2\}$
O^{priv_i}	$=$	$\{a_1\}$	$\{a_2\}$
$s_I^{\triangleright i}$	$=$	$\langle V_1, i_1\rangle, \langle V_3, u\rangle$	$\langle V_2, i_2\rangle, \langle V_3, u\rangle$
$s_\star^{\triangleright i}$	$=$	$\langle V_3, g\rangle$	$\langle V_3, g\rangle$

where the actions a_1, b_1 and a_2, b_2 are:

a	$\mathsf{pre}(a)$	$\mathsf{eff}(a)$	$\mathsf{cost}^1(a)$
a_1	$\langle V_1, \mathsf{i}_1\rangle$	$\langle V_1, \mathsf{p}_1\rangle$	$\mathsf{cost}^1(a_1) = 1$
b_1	$\langle V_1, \mathsf{p}_1\rangle$	$\langle V_1, \mathsf{i}_1\rangle, \langle V_3, g\rangle$	$\mathsf{cost}^1(b_1) = 2$
a_2	$\langle V_2, \mathsf{i}_2\rangle$	$\langle V_2, \mathsf{p}_2\rangle$	$\mathsf{cost}^2(a_2) = 1$
b_2	$\langle V_2, \mathsf{p}_2\rangle$	$\langle V_2, \mathsf{i}_2\rangle, \langle V_3, g\rangle$	$\mathsf{cost}^2(b_2) = 2$

In addition, the actions of projected problem $\Pi^{\triangleright 1}$ are $O^{\triangleright 1} = \{a_1^{\triangleright 1}, b_1^{\triangleright 1}, b_2^{\triangleright 1}\}$, where $a_1^{\triangleright 1}, b_1^{\triangleright 1}$ are unchanged and $b_2^{\triangleright 1}$:

a	$\mathsf{pre}(a)$	$\mathsf{eff}(a)$	$\mathsf{cost}^1(a)$
$b_2^{\triangleright 1}$	$\emptyset$	$\langle V_3, g\rangle$	$\mathsf{cost}^1(b_2^{\triangleright 1}) = 2$

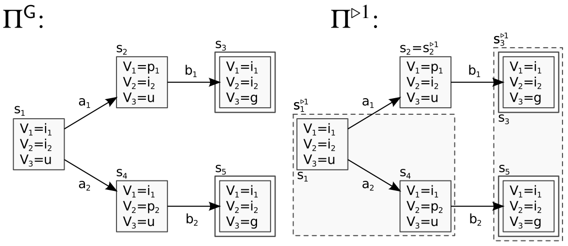

Figure 1: a) Transition system of the global problem Π^{G} respective to the example. b) Example transition system, 1-projection (abstraction).

Analogously, the actions of projected problem $\Pi^{\triangleright 2}$ are $O^{\triangleright 2} = \{a_2^{\triangleright 2}, b_2^{\triangleright 2}, b_1^{\triangleright 2}\}$, where $a_2^{\triangleright 2}, b_2^{\triangleright 2}$ are unchanged and $b_1^{\triangleright 2}$:

a	$\mathsf{pre}(a)$	$\mathsf{eff}(a)$	$\mathsf{cost}^2(a)$
$b_1^{\triangleright 2}$	$\emptyset$	$\langle V_3, g\rangle$	$\mathsf{cost}^2(b_1^{\triangleright 2}) = 2$

Obviously, a global solution to the problem is either (a_1, b_1) or (a_2, b_2), both of cost 3. The optimal solution of $\Pi^{\triangleright 1}$ is $(b_2^{\triangleright 1})$ with the cost of 2 and symmetrically for $\Pi^{\triangleright 2}$. Thus if we take the baseline approach and maximize the two optimal costs we obtain 2 which is a bound on the value any two admissible heuristics can give as a maximum of projected heuristics.

2.2 Transition Systems

The set $\mathcal{M}^{\triangleright}$ of all i-projected problems can be seen as a set of abstractions of the global problem Π^{G}. To do so, we first define the transition system of an MPT problem Π.

Definition 1. (Transition system) A transition system of a planning task Π is a tuple $\mathcal{T}(\Pi) = \langle S, L, T, s_I, S_\star \rangle$, where $S = \prod_{V \in \mathcal{V}} \mathsf{dom}(V)$ is a set of states, L is a set of transition labels corresponding to the actions in O and $T \subseteq S \times L \times S$ is a transition relation of Π s.t. $\langle s, a, s' \rangle \in T$ if $a \in O$ s.t. a is applicable in s and $s' = a \circ s$. A state-changing transition is $\langle s, a, s' \rangle \in T$ such that $s \neq s'$. The state $s_I \in S$ is the initial state and $S_\star$ is the set of all goal states (that is all states s s.t. $s_\star$ is consistent with s). The cost of a transition $\langle s, a, s' \rangle \in T$ is $\mathsf{cost}(a)$.

Next, we proceed with the definition of an abstraction.

Definition 2. (Abstraction) Let $\mathcal{T} = \langle S, L, T, s_I, S_\star \rangle$ and $\mathcal{T}' = \langle S', L', T', s'_I, S'_\star \rangle$ be transition systems with the same label set $L = L'$ and let $\sigma : S, S'$. We say that T' is an abstraction of T with abstraction function (mapping) σ if

- $s'_I = \sigma(s_I)$,
- for all $s \in S_\star$ also $\sigma(s) \in S'_\star$, and

- for all $\langle s,a,s'\rangle \in T$, $\langle \sigma(s),a,\sigma(s')\rangle \in T'$.

To conclude this section, we show that an i-projection is an abstraction.

Theorem 3. *(Projection is an abstraction) Let $\mathcal{T}(\Pi^{\mathsf{G}}) = \langle S^{\mathsf{G}}, \bigcup_{i\in 1..n} O^i, T^{\mathsf{G}}, s_I, S_\star\rangle$ be the transition system of the global problem Π^{G} and $\mathcal{T}(\Pi^{\triangleright i}) = \langle S^{\triangleright i}, O^{\triangleright i}, T^{\triangleright i}, s_I^i, S_\star^i\rangle$* the transition system of the i-projected problem $\Pi^{\triangleright i}$. Then $\mathcal{T}(\Pi^{\triangleright i})$ is an abstraction of $\mathcal{T}(\Pi^{\mathsf{G}})$ with respect to the state-changing transitions.

Proof. We define an abstraction mapping $\sigma^{\triangleright i} : S^{\mathsf{G}}, S^{\triangleright i}$ such that for a state $s \in S^{\mathsf{G}}$ we define $\sigma^{\triangleright i}(s)$ as a restriction of s to the variables in $\mathcal{V}^i$. Then from definition, $\sigma^{\triangleright i}(s) = s^{\triangleright i}$. From definition also $s_I^{\triangleright i} = \sigma^{\triangleright i}(s_I)$. If $s \in S_\star$ then $s_\star$ is compatible with s, if both are restricted to $\mathcal{V}^i$, the compatibility is not violated and thus $\sigma^{\triangleright i}(s) \in S_\star^i$.

For each action $a \in O^i$ and each transition $\langle s,a,s'\rangle \in T^{\mathsf{G}}$ there is a transition $\langle s^{\triangleright i}, a^{\triangleright i}, s'^{\triangleright i}\rangle \in T^{\triangleright i}$ as $a^{\triangleright i} = a$. For $j \neq i$ and for each action $a' \in O^{\mathsf{pub}_j}$ and each transition $\langle t,a',t'\rangle \in T^{\mathsf{G}}$, there is a transition $\langle t^{\triangleright i}, a'^{\triangleright i}, t'^{\triangleright i}\rangle \in T^{\triangleright i}$ as $\mathsf{pre}(a'^{\triangleright i})$ is $\mathsf{pre}(a')$ restricted to $\mathcal{V}^i$ and $t^{\triangleright i}$ is t restricted to $\mathcal{V}^i$ (the same goes for $\mathsf{eff}(a^{\triangleright i})$. For each action $a'' \in O^{\mathsf{priv}_j}$ and each transition $\langle u,a',u'\rangle \in T^{\mathsf{G}}$, there is no transition $\langle u^{\triangleright i}, a''^{\triangleright i}, u'^{\triangleright i}\rangle \in T^{\triangleright i}$, but as both $\mathsf{pre}(a'')$ and $\mathsf{eff}(a'')$ are defined only over $\mathcal{V}^{\mathsf{priv}_j}$, $u^{\triangleright i} = u'^{\triangleright i}$ and thus the missing transition $\langle u^{\triangleright i}, a''^{\triangleright i}, u'^{\triangleright i}\rangle \in T^{\triangleright i}$ is a loop. □

The missing loops never influence the shortest path and thus can be ignored (or added at will).

3 COST PARTITIONING

In this section, we describe the idea of cost partitioning (Katz and Domshlak, 2010) as used in classical planning and define a novel notion of multi-agent cost partitioning. We restrict ourselves to the non-negative cost partitioning, where the costs of actions are not allowed to be less than 0, but all notions and techniques generalize to the case of general cost partitioning without such restriction.

Definition 4. (Cost partitioning). Let Π be a planning task with operators O and cost function cost. A cost partitioning for Π is a tuple $\mathsf{cp} = \langle \mathsf{cp}_1, ..., \mathsf{cp}_k\rangle$ where $\mathsf{cp}_l : O \to \mathbb{R}_0^+$ for $1 \leq l \leq k$ and $\sum_{l=1}^{k} \mathsf{cp}_l(o) \leq \mathsf{cost}(o)$ for all $o \in O$.

As shown in (Katz and Domshlak, 2010), a sum of admissible heuristics computed on the cost partitioned problem is also admissible, formally

Proposition 5. *(Katz and Domshlak 2010). Let Π be a planning task, let $h_1, ..., h_k$ be admissible heuristics for Π, and let $\mathsf{cp} = \langle \mathsf{cp}_1, ..., \mathsf{cp}_k\rangle$ be a cost partitioning for Π. Then $h_{\mathsf{cp}} = \sum_{l=1}^{k} h_l(s)$ where each h_l is computed with cp_l is an admissible heuristic estimate for a state s.*

Based on the particular cost partitioning cp, the heuristic estimate can have varying quality. By optimal cost partitioning (OCP) we mean a cost partitioning which maximizes h_{cp}. Now we proceed with the definition of a multi-agent variant of cost partitioning, which differs in that the partitions are defined apriori by the set of the i-projected problems.

Definition 6. (Multi-agent cost partitioning). Let $\mathcal{M}^{\triangleright} = \{\Pi^{\triangleright i}\}_{i=1}^{n}$ be the set of all i-projected problems with respective cost functions $\mathsf{cost}^{\triangleright i}$. A multi-agent cost partitioning for $\mathcal{M}^{\triangleright}$ is a tuple of functions $\mathsf{cp} = \langle \mathsf{cp}_1, ..., \mathsf{cp}_n\rangle$ where $\mathsf{cp}_i : O^{\triangleright i} \to \mathbb{R}_0^+$. For $1 \leq i \leq n$ and for each $o \in O^{\mathsf{G}}$ holds $\sum_{i=1}^{n} \mathsf{cp}_i(o^{\triangleright i}) \leq \mathsf{cost}^j(o)$ where α_j is the owner of o, that is $o \in O^j$.

Theorem 7. *Let $\mathcal{M}^{\triangleright} = \{\Pi^{\triangleright i}\}_{i=1}^{n}$ be the set of all i-projected problems, Π^{G} the global problem respective to $\mathcal{M}$ and cp a multi-agent cost partitioning for $\mathcal{M}^{\triangleright}$. Then cp is a cost partitioning for Π^{G}.*

Proof. The theorem follows from Definition 4, Definition 6 for all public actions and from setting $o^{\triangleright i} = \varepsilon$ for all $o \in O^{\mathsf{priv}_j}$ s.t. $j \neq i$. As $\mathsf{cost}^{\triangleright i}(o^{\triangleright i}) = \mathsf{cost}^{\triangleright i}(\varepsilon) = 0$ and $\mathsf{cost}^{\triangleright j}(o^{\triangleright j}) = \mathsf{cost}^j(o)$, the cost partitioning property $\sum_{i=1}^{n} \mathsf{cp}_i(o^{\triangleright i}) \leq \mathsf{cost}^j(o)$ holds also for private operators. □

Thanks to Theorem 7 we can apply the Proposition 5 also in the multi-agent setting using a multi-agent cost partitioning. Thus, each agent α_i can compute its part of the heuristic locally on $\Pi^{\triangleright i}$ using cp_i instead of cost^i as the cost function. To obtain the global heuristic, the individual parts can be simply summed

$$h_{\mathsf{cp}}(s) = \sum_{i=1}^{n} h_{\mathsf{cp}_i}^{\triangleright i}(s^{\triangleright i}) \tag{1}$$

where $h_{\mathsf{cp}_i}^{\triangleright i}$ is an i-projected heuristic computed on $\Pi^{\triangleright i}$ using cp_i. We contrast this approach to a baseline solution which is the current state of the art. The baseline combines individual projected heuristics by taking the maximum, formally

$$h_{\max}(s) = \max_{1\leq i\leq n} h^{\triangleright i}(s^{\triangleright i}) \tag{2}$$

where $h^{\triangleright i}$ is any (admissible) heuristic computed on $\Pi^{\triangleright i}$ using the original cost^i.

4 COMPUTING MULTI-AGENT COST PARTITIONING

To compute the optimal cost partitioning (OCP) for i-projections, based on Theorems 3 and 7 we can readily apply the results from classical planning. We adopt the computation of optimal cost partitioning for abstractions (Pommerening et al., 2014b) and introduce a novel cost partitioning based on the potential heuristic linear program formulation (Pommerening et al., 2015) which we later show to be well suited to the problem at hand.

Note that in all cost partitionings private actions are partitioned implicitly as only their respective owners are aware of them, formally:

$$\mathsf{cp}_j(a^{\triangleright j}) = \begin{cases} \mathsf{cost}^i(a^{\triangleright i}) & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

The very baseline is the uniform cost partitioning, where

$$\mathsf{cp}_j(a^{\triangleright j}) = \frac{\mathsf{cost}^i(a^{\triangleright i})}{n} \tag{3}$$

for each action $a \in O^{\mathsf{pub}_i}$ and each agent $\alpha_j \in \mathcal{A}$.

Example. On the running example, the uniform CP results in $\mathsf{cp}_1(b_1^{\triangleright 1}) = 1, \mathsf{cp}_1(b_2^{\triangleright 1}) = 1$ and $\mathsf{cp}_2(b_1^{\triangleright 2}) = 1, \mathsf{cp}_2(b_2^{\triangleright 2}) = 1$. The sum of optimal costs computed on such cost partitioning is 2.

4.1 Optimal Cost Partitioning on Abstractions

The idea behind the following linear program (LP) is to encode the abstract transition systems and possible shortest paths in it. The LP variables used for each $\alpha_i \in \mathcal{A}$ are $\bar{h}^{\triangleright i}$ encoding the i-projected heuristic value (given the cost partitioning), $\bar{s}'^{\triangleright i}$ representing the cost of shortest path from a state s (or actually $s^{\triangleright i}$) to $s'^{\triangleright i}$ in the i-projected problem given the cost partitioning and $\bar{a}^{\triangleright i}$ representing the cost partitioned cost of action $a^{\triangleright i} \in O^{\triangleright i}$. The LP is formulated as follows:

Maximize $\sum_{i=1}^{n} \bar{h}^{\triangleright i}$ subject to

$$\begin{array}{rcll} \bar{s}'^{\triangleright i} & = & 0 & \text{for all}\, s'^{\triangleright i} = s^{\triangleright i} \\ \bar{s}''^{\triangleright i} & \leq & \bar{s}'^{\triangleright i} + \bar{a}^{\triangleright i} & \text{for all}\, \left\langle s'^{\triangleright i}, a^{\triangleright i}, s''^{\triangleright i} \right\rangle \in T^{\triangleright i} \\ \bar{h}^{\triangleright i} & \leq & \bar{s}'^{\triangleright i} & \text{for all}\, \bar{s}'^{\triangleright i} \in s^i_\star \\ \sum_{j=1}^{n} \bar{a}^{\triangleright j} & \leq & \mathsf{cost}^i(a) & \text{for all}\, a \in O^{\mathsf{pub}_i} \\ \bar{a}^{\triangleright i} & \leq & \mathsf{cost}^i(a) & \text{for all}\, a \in O^{\mathsf{priv}_i} \end{array} \tag{4}$$

where the first set of constraints sets all states equal (in the i-projection) with the current state s to have zero cost of shortest path. The second set of constraints encode the actual (abstracted) transitions and their costs (transitions where $s'^{\triangleright i} = s''^{\triangleright i}$ can be ignored), the third set of constraints places an upper bound on the actual heuristic estimate to keep it admissible. The fourth and fifth sets of constraints represent the cost partitioning of public and private actions respectively. As already mentioned, private actions of agent α_i always occur only as i-projections and are not partitioned (i.e. any other projection of such action has the cost of 0).

Example. Let us show how the optimal cost partitioning is computed on the running example. The global transition system is shown in Figure 1 a) and the transition system projected to agent α_1 in Figure 1 b) (transition system projected to α_2 is symmetrical). The LP is constructed according to Equation 4 and the solution gives $h^{\triangleright 1} + h^{\triangleright 2} = 3$ as the value of the objective function and $\bar{b}_1^{\triangleright 1} = 1, \bar{b}_1^{\triangleright 2} = 1, \bar{b}_2^{\triangleright 1} = 2, \bar{b}_2^{\triangleright 2} = 0$ as the values of (relevant) LP variables. The values directly give the cost partitioning. When applied, the optimal solutions of $\Pi^{\triangleright 1}$ and $\Pi^{\triangleright 2}$ has the cost of 1 and 2 respectively, which is the maximal value so that the sum does not violate admissibility.

In contrast to the use in classical planning, we intend to compute the cost partitioning LP only once at the beginning of the planning process. Obviously, this results in a possibly sub-optimal cost partitioning for other states than the initial one.

Unfortunately, even computing such OCP once may be intractable in general, as the i-projected problems may be as large and as hard as the global problem, e.g., in a scenario where all (or most of) actions and variables are public. The optimal cost partitioning can be approximated by first constructing smaller abstractions out of the i-projections using some standard technique such as Merge & Shrink (Helmert et al., 2007).

4.2 Cost Partitioning based on Potential Heuristic

Potential heuristics are a family of admissible heuristics introduced in (Pommerening et al., 2015). As shown in (Štolba et al., 2016a), the potential heuristics are very well suited for distributed heuristic computation. In this work, we apply the LP used to compute the potentials to derive a multi-agent cost partitioning.

We first briefly describe the original centralized version of the potential heuristic (denoted as h_{pot}) and the LP used to compute it. A potential heuristic associates a numerical potential with each fact. The potential heuristic for a state s is simply a sum of potentials of the facts in s. The potentials can be determined as a solution to a linear program, a detailed formulation is described in (Pommerening et al., 2014a). The objective function of the LP is simply the sum of potentials for a state (or average for a set of states).

The simplest variant is to use the initial state s_I as the optimization target.

For a partial variable assignment p, let $\mathsf{maxpot}(V,p)$ denote the maximal potential that a state consistent with p can have for variable V, formally:

$$\mathsf{maxpot}(V,p)=\begin{cases}\mathsf{pot}(\langle V,p[V]\rangle) & \text{if} V\in\mathsf{vars}(p)\\ \max\limits_{v\in\mathsf{dom}(V)}\mathsf{pot}(\langle V,v\rangle) & \text{otherwise}\end{cases}$$

The LP will have a potential LP-variable $\mathsf{pot}(\langle V,v\rangle)$ for each fact (that is each possible assignment to each variable) and a maximum potential LP-variable maxpot_V for each variable in $\mathcal{V}$. The constraints ensuring the maximum potential property are simply

$$\mathsf{pot}(\langle V,v\rangle)\leq\mathsf{maxpot}_V \tag{5}$$

for all variables V and their values $v\in\mathsf{dom}(V)$. To ensure goal-awareness of the heuristic, i.e., $h_{\mathsf{pot}}(s)\leq 0$ for all goal states s, we add the following constraint

$$\sum_{V\in\mathcal{V}}\mathsf{maxpot}(V,s_\star)\leq 0 \tag{6}$$

restricting the heuristic of any goal state to be less or equal to 0. The final set of constraints ensures consistency. A consistent heuristic is such h that for every two states s,s' and all operators s.t. $s'=s\circ o$ holds $h(s)\leq h(s')+\mathsf{cost}(o)$. Consistency together with the goal-awareness implies admissibility. For each operator o in a set of operators O we add the following constraint

$$\sum_{V\in\mathsf{vars}(\mathsf{eff}(o))}(\mathsf{maxpot}(V,\mathsf{pre}(o))-\mathsf{pot}(\langle V,\mathsf{eff}(o)[V]\rangle)) \leq\mathsf{cost}(o) \tag{7}$$

The optimization function of the LP can be set to the sum of potentials in the initial state. A solution of the LP yields the values for potentials which are then used in the heuristic computation.

Example. Let us consider the running example. The consistency constraint of the potential heuristic LP constructed for the public action b_2 is

$$\mathsf{pot}(\langle V_2,p_2\rangle)-\mathsf{pot}(\langle V_2,i_2\rangle)+\mathsf{maxpot}_{V_3}-\mathsf{pot}(\langle V_3,g\rangle)\leq\mathsf{cost}^2(b_2)$$

where $\mathsf{cost}^2(b_2)=2$. For the variable V_2 we use the potential for the precondition and the effect and for V_3 we use the maxpot_{V_3} variable for the precondition as V_3 has no value in the precondition of b_2.

We can simply obtain a cost partitioning LP by replacing the action costs with variables, concatenating the respective LPs for each of the agent problems and adding the cost partitioning constraints. There are separate LP variables even for the potentials of public variables for each of the agents. Let $o\in O^{\mathsf{pub}_i}$ be a public operator of α_i, the consistency constraints from Equation 7 for operator o are formulated as

$$\sum_{V\in\mathsf{vars}(\mathsf{eff}(o))}(\mathsf{maxpot}(V,\mathsf{pre}(o))^{\rhd i}-\mathsf{pot}(\langle V,\mathsf{eff}(o)[V]\rangle)^{\rhd i})\leq\bar{o}^{\rhd i} \tag{8}$$

$$\sum_{V\in\mathsf{vars}(\mathsf{eff}(o^{\rhd j}))}(\mathsf{maxpot}(V,\mathsf{pre}(o^{\rhd j}))^{\rhd j}-\mathsf{pot}(\left\langle V,\mathsf{eff}(o^{\rhd j})[V]\right\rangle)^{\rhd j})\leq\bar{o}^{\rhd j}\quad\forall j\neq i \tag{9}$$

$$\sum_{k=1}^{n}\bar{o}^{\rhd k}\leq\mathsf{cost}^i(o) \tag{10}$$

where $\mathsf{maxpot}(V,v)^{\rhd k}$ and $\mathsf{pot}(\langle V,v\rangle)^{\rhd i}$ represent the LP variables respective to agent α_k. Note that in the case of projected operators $o^{\rhd j}$, the set $\mathsf{vars}(\mathsf{eff}(o^{\rhd j}))$ contains only public variables. The cost partitioning LP can also be seen as a set of n individual potential heuristic LPs which are interconnected only by the cost partitioning variables $\bar{o}^{\rhd k}$ and the respective CP constraint. Also, the optimization function can be constructed simply as a sum of the individual optimization functions.

A significant advantage of the potential-based CP over the abstraction-based OCP is that it is computationally tractable as the whole transition system does not have to be constructed. Moreover, the distributed LP computation techniques from (Štolba et al., 2016a) can be used in the case of the potential-based CP as well.

Example. The action b_2 will then be represented by two consistency constraints, one for b_2 in the context of Π^1 and one for $b_2^{\rhd 2}$ in the context of $\Pi^{\rhd 2}$. The constraints for b_2 (including the cost partitioning constraint) are as follows.

$$\begin{aligned}&\mathsf{pot}(\langle V_2,p_2\rangle)^{\rhd 2}-\mathsf{pot}(\langle V_2,i_2\rangle)^{\rhd 2}+\\&\quad\mathsf{maxpot}_{V_3}^{\rhd 2}-\mathsf{pot}(\langle V_3,g\rangle)^{\rhd 2}\leq\bar{b}_2^{\rhd 2}\\&\quad\mathsf{maxpot}_{V_3}^{1}-\mathsf{pot}(\langle V_3,g\rangle)^{\rhd 1}\leq\bar{b}_2^{\rhd 1}\\&\quad\bar{b}_2^{\rhd 1}+\bar{b}_2^{\rhd 2}\leq\mathsf{cost}^2(b_2)\end{aligned}$$

where $\mathsf{cost}^2(b_2)=2$. Other constraints are formulated analogously. There are multiple possibilities for the optimization function, if we base the function on the initial state, we obtain the following one:

$$\begin{aligned}&\mathsf{Maximize:}\\&\mathsf{pot}(\langle V_1,i_1\rangle)^{\rhd 1}+\mathsf{pot}(\langle V_2,i_2\rangle)^{\rhd 2}+\\&\quad\mathsf{pot}(\langle V_3,u\rangle)^{\rhd 1}+\mathsf{pot}(\langle V_3,u\rangle)^{\rhd 2}\end{aligned}$$

The resulting cost partitioning is $\bar{b}_1^{\rhd 1}=1,\bar{b}_1^{\rhd 2}=1,\bar{b}_2^{\rhd 1}=2,\bar{b}_2^{\rhd 2}=0$ which gives $h^{\rhd 1}+h^{\rhd 2}=3$, that is, the same value as the optimal cost partitioning based on abstractions.

5 SEARCH WITH AGENT-ADDITIVE HEURISTICS

In this work, we aim to provide a general technique for additive heuristic computation in multi-agent planning. By additive we mean that each part of the heuristic can be computed by each respective agent separately an then added together.

Definition 8. (Agent-additive heuristic) A global heuristic h estimating the global problem Π^{G} is agent-additive iff for any agent $\alpha_i \in \mathcal{A}$ it can be represented as

$$h(s)=h^{\mathsf{pub}}(s^{\rhd})+\sum_{\alpha_j\in\mathcal{A}}h^j(s^{\rhd j})$$

where h^{pub} is a heuristic computed on the public projection problem $\Pi^{\rhd}$ and h^j is a heuristic computed on the j-projected problem $\Pi^{\rhd j}$.

A heuristic is agent-additive even without the public part, that is, if $h^{\mathsf{pub}}(s^{\rhd})=0$ for all states, which is the case of the heuristic computed on multi-agent cost partitioning defined in Equation 1.

Theorem 9. *Let $\mathcal{M}^{\rhd}=\{\Pi^{\rhd i}\}_{i=1}^n$ be the set of all i-projected problems, Π^{G} the global problem respective to $\mathcal{M}$ and cp a multi-agent cost partitioning for $\mathcal{M}^{\rhd}$. Then the heuristic $h_{\mathsf{cp}}(s)$ (Equation 1) is agent-additive (Definition 8).*

Proof. Follows trivially from Definition 8 by having the public part equal to zero, that is, $h^{\mathsf{pub}}(s^{\rhd})$ for all i and all states. □

In the rest of this section, we show how the agent-additive property can be utilized in the search. The principle of the multi-agent heuristic search presented here is based on the MAD-A* algorithm (Multi-Agent Distributed A*) (Nissim and Brafman, 2012). We first briefly summarize the main principles. The MAD-A* algorithm is a simple extension of classical A*. The agents search in parallel, possibly in a distributed setting (i.e., communicating over a network). Each agent $\alpha_i \in \mathcal{A}$ searches using its operators from O^i and if a state s is expanded using a public operator $o \in O^{\mathsf{pub}_i}$, the resulting state s' is sent to other agents (the agents may be filtered in order to send the state only to the relevant ones). When some other agent α_j receives the state s', s' is added to the OPEN list of α_j and expanded normally when due. The original MAD-A* uses only projected heuristics computed on $\Pi^{\rhd i}$. Each state sent by α_i is also accompanied with its i-projected heuristic estimate and when received, the receiving agent α_j computes the j-projected heuristic estimate of the received state s' and takes $h(s)=\max(h^{\rhd i}(s^{\rhd i}),h^{\rhd j}(s^{\rhd j}))$.

Let us now consider how can the agent-additive heuristic be utilized in the search to reduce heuristic computation and communication. In order to do so, we first state the following two propositions.

Proposition 10. *Let $\mathcal{M}=\{\Pi^i\}_{i=1}^n$ be a multi-agent problem and let $h(s)=h^{\mathsf{pub}}(s^{\rhd})+\sum_{\alpha_j\in\mathcal{A}}h^i(s^{\rhd i})$ be an agent-additive heuristic. Let s and s' be two states where s' is created from s by the application of a private operator $o \in O^{\mathsf{priv}_j}$ of some agent j. Then for all h^j such that $j \neq i$ holds $h^j(s^{\rhd j})=h^j(s'^{\rhd j})$ and*

$$h(s')=h(s)-h^{\mathsf{pub}}(s^{\rhd})-h^i(s^{\rhd i})+h^{\mathsf{pub}}(s'^{\rhd})+h^i(s'^{\rhd i}) \quad (11)$$

Proof. As $o \in O^{\mathsf{priv}_i}$ the states s,s' differ only in variables private to agent i and thus $s^{\rhd j}=s'^{\rhd j}$ and consequently $h^j(s^{\rhd j})=h^j(s'^{\rhd j})$ for all $j \neq i$. Equation 11 follows directly from the fact that from the point of view of the agent i, the value of the private parts of the agent-additive heuristic of all other agents can be expressed as $\sum_{\alpha_j\in\mathcal{A}\setminus\{\alpha_i\}}h^j(s^{\rhd j})=h(s)-h^{\mathsf{pub}}(s^{\rhd})-h^i(s^{\rhd i})$. □

This means, that the heuristic estimate of a state s' can be easily determined from the heuristic estimate of its predecessor s if s' was obtained from s by the application of a private action. When a state is received from some other agent j, it is accompanied with its global heuristic estimate computed by agent j. When a state s is expanded by agent i with a private action, the heuristic estimate of its successor s' can be computed using Equation 11. In order to avoid excessive heuristic computations, the values of $h^{\mathsf{pub}}(s^{\rhd})$ and $h^i(s^{\rhd i})$ can be cached when first computed.

Proposition 11. *Let $\mathcal{M}=\{\Pi^i\}_{i=1}^n$ be a multi-agent problem and let $h(s)=h^{\mathsf{pub}}(s^{\rhd})+\sum_{\alpha_j\in\mathcal{A}}h^i(s^{\rhd i})$ be an agent-additive heuristic. Let s and s' be two states such that for some agent j holds $s^{\rhd j}=s'^{\rhd j}$. Then $h^j(s^{\rhd j})=h^j(s'^{\rhd j})$.*

Proof. Holds trivially. □

Proposition 11 means that from the perspective of agent i, the states s,s' differ only in the private parts of other agents. This information can be again used to reduce the heuristic computation by caching the values of $h^i(s^{\rhd i})$. If the computation of $h^i(s'^{\rhd i})$ is requested for some state s' such that $s^{\rhd j} = s'^{\rhd j}$, the cached value can be returned directly.

6 EVALUATION

We have evaluated the proposed approach on the benchmark set of the CoDMAP'15 (Komenda et al., 2016) competition. In the evaluation we focus on three key metrics: i) the heuristic value in the initial state, ii) the number of expanded states, and iii) the number of problems solved in time limit of 30 minutes (coverage). The proposed methods were implemented in the MA-Plan planner (Fišer et al., 2015) and evaluated on the LM-Cut heuristic (Helmert and Domshlak, 2009). The configurations we have evaluated are the following:

max is the baseline solution where the heuristic is computed on the projected problems without any cost partitioning. The resulting heuristic is computed using maximum (Equation 1). We have also evaluated **proj** which is the classical projection heuristic computed by a single agent only.

uni is the uniform baseline cost partitioning (Equation 3).

abs-N is the abstraction-based optimal cost partitioning computed using the LP in Equation 4 where N denotes the number of states of the abstraction of each agent. The abstractions are computed using the Merge&Shrink heuristic (Helmert et al., 2007) implemented in Fast-Downward (Helmert, 2006).

pot is the potential-based cost partitioning computed using the LP in Equation 10. The implementation is based on the distributed potential heuristic LP computation (Štolba et al., 2016a).

6.1 Heuristic Quality

In this section, we focus on the quality of the heuristic computed using the cost partitioning (CP) and computed as a maximum of projections. First, we compare the heuristic values computed for the initial state by each of the configuration. Note that the CPs are optimized for the initial state and thus may give worse heuristic estimates for further states during the search.

Figure 2 shows the comparison of heuristic values for the best performing configurations. In both cases, we can see that there are some problems where the CP gives lower estimates and some cases where it gives higher. The most prominent example is the elevators domain where both CPs outperform the baseline.

Table 1: Numbers of problems solved (coverage).

domain	**abs 1000**	**abs 10k**	**proj**	**max**	**uni**	**pot**
blocksworld	2	2	0	**3**	2	**3**
depot	1	**3**	**3**	0	1	1
driverlog	11	11	**14**	12	**14**	12
elevators	2	1	5	5	6	**9**
logistics	4	6	5	**8**	**8**	7
rovers	**5**	2	2	4	4	**5**
satellites	6	2	1	3	5	**7**
sokoban	6	5	**13**	10	8	8
taxi	**3**	1	**3**	**3**	1	**3**
wireless	0	0	**1**	**1**	0	0
woodwork.	**1**	**1**	**1**	0	**1**	0
zenotravel	7	6	6	**9**	8	**9**
Σ	48	40	54	58	58	**64**

Let us now focus on the actual heuristic guidance during the search, that is, the number of expanded states shown in Figure 3. The figure shows, that in many smaller problems, the abstraction-based CP provides worse guidance. In the taxi domain and a couple of other larger problems, the performance is improved. The potential-based CP exhibits a similar pattern.

6.2 Coverage

In this section, we focus on the actual performance of the MAD-A* algorithm together with the proposed heuristics and the search improvements described in this work. In order to perform this evaluation, we have replicated the configuration of the distributed track of the CoDMAP'15 (Komenda et al., 2016) competition, where each agent runs on a dedicated machine with 4 cores on 3.2GHz and 8GB[1] RAM. The agents communicated over TCP-IP on a local area network. The main difference between our setting and the competition is a different network topology, in our case, there were multiple switches between some nodes which could have negatively influenced the performance.

The results in Table 1 show a number of interesting points. First, the classical projection (proj) and the maximum of projections (max) are on par overall but perform differently on some domains, e.g., logistics, where the max heuristic benefits from the additional information. Second, the abstraction-based CPs (abs) perform badly. This is because of the time and especially memory requirements of the abstraction computation. This may possibly be improved by a future work on the CP computation. Finally, the best performance is obtained by the potential-based

[1]Note, that the Fast-Downward planner used to compute the abstractions was limited to 3GB due to the 32-bit architecture (1GB is restricted for the kernel).

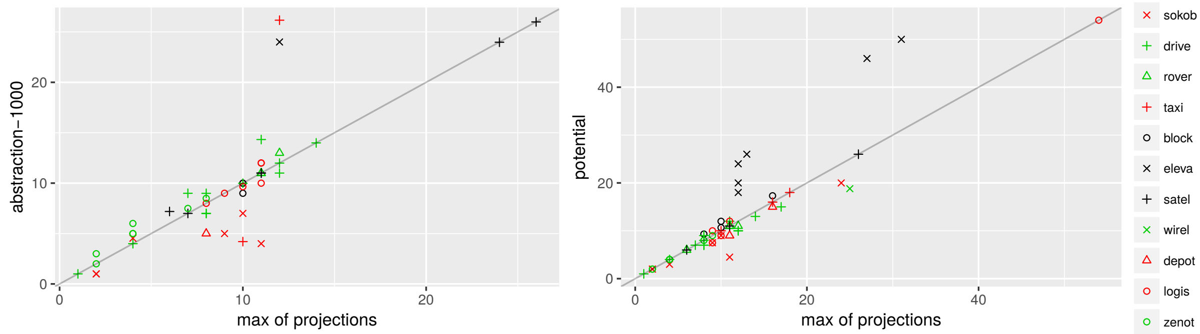

Figure 2: Initial state heuristics of the abstraction-based and potential-based cost partitionings compared with the maximum of projections.

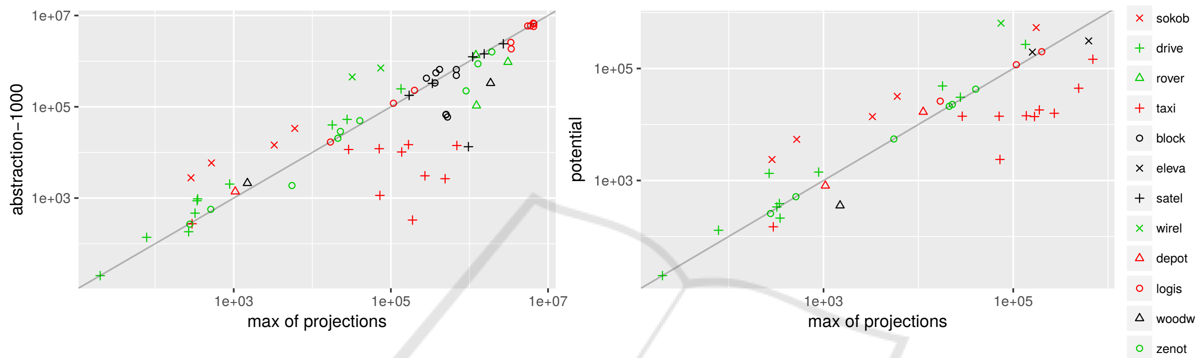

Figure 3: Expanded states on solved problems of the abstraction-based and potential-based cost partitionings compared with the maximum of projections (log scale).

CP (pot) which is relatively fast to compute and improves heuristic guidance on a significant number of problems.

7 DISCUSSION OF PRIVACY

Privacy is a crucial issue in multi-agent planning, here we discuss a number of issues concerning privacy in the context of multi-agent cost partitioning.

Let us first focus on the heuristic computation itself, assuming we already have a cost-partitioning. The authors in (Tožička et al., 2017) have shown that a search-based algorithm such as MAFS or MAD-A* can never preserve privacy fully. Moreover, the authors in (Štolba et al., 2016b) have proposed a method for measuring the amount of leaked private information. We argue that the heuristic h_{cp} based on the cost-partitioning does not leak more information than the baseline h_{max}, the classical projection-based heuristic, or any distributed heuristic (with respect to the heuristic values) as the sum of the individual h_{cp_i} can be computed using a secure sum algorithm, e.g., (Sheikh et al., 2010) (causing additional computation).

Naturally, the computation of the cost partitioning may leak some private information as well, except for the most simple variants such as the uniform CP. The potential-based CP is computed using an LP very similar to the original potential heuristic LP. A secure variant of the potential LP computation was proposed in (Štolba et al., 2016a), the same technique can be used in our case, thus avoiding the information leakage. We assume that a similar technique would be applicable to the abstraction-based CP LP computation.

The last source of private information leakage is the cost partitioning itself, that is, the new costs of the public action which somehow reflect the structure of the private parts of the problem. We leave the analysis of how much and what information can be learned from the cost partitioning for future work.

8 CONCLUSION AND FUTURE WORK

In this work, we have presented a novel general technique to compute distributed admissible heuristics for multi-agent planning based on the principle of cost partitioning. We have shown that this approach is promising and improves over the baseline projected heuristics.

A promising future work is the development of a cost partitioning more specific to the multi-agent planning.

ACKNOWLEDGEMENTS

This research was supported by the Czech Science Foundation (grant no. 18-24965Y). The authors acknowledge the support of the OP VVV MEYS funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics".

REFERENCES

Brafman, R. I. (2015). A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pages 1530–1536.

Brafman, R. I. and Domshlak, C. (2008). From one to many: Planning for loosely coupled multi-agent systems. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS'08)*, pages 28–35.

Dimopoulos, Y., Hashmi, M. A., and Moraitis, P. (2012). μ-satplan: Multi-agent planning as satisfiability. *Knowledge-Based Systems*, 29:54–62.

Fikes, R. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI'71)*, pages 608–620.

Fišer, D., Štolba, M., and Komenda, A. (2015). MAPlan. In *Proceedings of the Competition of Distributed and Multi-Agent Planners (CoDMAP-15)*, pages 8–10.

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

Helmert, M. and Domshlak, C. (2009). Landmarks, critical paths and abstractions: What's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 162–169.

Helmert, M., Haslum, P., and Hoffmann, J. (2007). Flexible abstraction heuristics for optimal sequential planning. In *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS'07)*, pages 176–183.

Jezequel, L. and Fabre, E. (2012). A#: A distributed version of A* for factored planning. In *Proceedings of the 51th IEEE Conference on Decision and Control, (CDC'12)*, pages 7377–7382.

Katz, M. and Domshlak, C. (2010). Implicit abstraction heuristics. *Journal of Artificial Intelligence Research*, pages 51–126.

Komenda, A., Stolba, M., and Kovacs, D. L. (2016). The international competition of distributed and multiagent planners (CoDMAP). *AI Magazine*, 37(3):109–115.

Maliah, S., Brafman, R. I., and Shani, G. (2016). Privacy preserving LAMA. In *Proceedings of the 4th Workshop on Distributed and Multi-Agent Planning, DMAP–ICAPS'16*, pages 1–5.

Maliah, S., Shani, G., and Stern, R. (2015). Privacy preserving pattern databases. In *Proceedings of the 3rd Distributed and Multiagent Planning (DMAP) Workshop of ICAPS'15*, pages 9–17.

Nissim, R. and Brafman, R. I. (2012). Multi-agent A* for parallel and distributed systems. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'12)*, pages 1265–1266.

Nissim, R. and Brafman, R. I. (2014). Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research*, 51:293–332.

Pommerening, F., Helmert, M., Röger, G., and Seipp, J. (2014a). From non-negative to general operator cost partitioning: Proof details. *Technical Report CS-2014-005, University of Basel, Department of Mathematics and Computer Science*.

Pommerening, F., Helmert, M., Röger, G., and Seipp, J. (2015). From non-negative to general operator cost partitioning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 3335–3341.

Pommerening, F., Röger, G., Helmert, M., and Bonet, B. (2014b). LP-Based heuristics for cost-optimal planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 226–234.

Sheikh, R., Kumar, B., and Mishra, D. K. (2010). A distributed k-secure sum protocol for secure multi-party computations. *arXiv preprint arXiv:1003.4071*.

Štolba, M., Fišer, D., and Komenda, A. (2015). Admissible landmark heuristic for multi-agent planning. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, pages 211–219.

Štolba, M., Fišer, D., and Komenda, A. (2016a). Potential heuristics for multi-agent planning. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling, ICAPS'16*, pages 308–316.

Štolba, M., Tožička, J., and Komenda, A. (2016b). Secure multi-agent planning. In *Proceedings of the 1st International Workshop on AI for Privacy and Security*, pages 11:1–11:8. ACM.

Torreño, A., Onaindia, E., and Sapena, O. (2014). FMAP: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606–626.

Tožička, J., Jakubuv, J., and Komenda, A. (2014). Generating multi-agent plans by distributed intersection of Finite State Machines. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, pages 1111–1112.

Tožička, J., Štolba, M., and Komenda, A. (2017). The limits of strong privacy preserving multi-agent planning. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling, ICAPS'17*.