Potential Heuristics for Multi-Agent Planning

Conference Paper · June 2016

CITATIONS
3

READS
110

3 authors:

Daniel Fišer
16 PUBLICATIONS **279** CITATIONS
SEE PROFILE

Antonín Komenda
Czech Technical University in Prague
75 PUBLICATIONS **503** CITATIONS
SEE PROFILE

Michal Štolba
Czech Technical University in Prague
25 PUBLICATIONS **262** CITATIONS
SEE PROFILE

Potential Heuristics for Multi-Agent Planning

Michal Štolba and **Daniel Fišer** and **Antonín Komenda**
{stolba,fiser,komenda}@agents.fel.cvut.cz
Department of Computer Science, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic

Abstract

Distributed heuristic search is a well established technique for multi-agent planning. It has been shown that distributed heuristics may crucially improve the search guidance, but are costly in terms of communication and computation time. One solution is to compute a heuristic additively, in the sense that each agent can compute its part of the heuristic independently and obtain a complete heuristic estimate by summing up the individual parts. In this paper, we show that the recently published potential heuristic is a good candidate for such heuristic, moreover admissible. We also demonstrate how the multi-agent distributed A* search can be modified in order to benefit from such additive heuristic. The modified search equipped with a distributed potential heuristic outperforms the state of the art.

Introduction

If planning is to be used in large-scale personal, corporate or military applications, multiple independent entities will need to cooperate in the plan synthesis. As witnessed in many other applications, such independent entities may have serious concern in protecting the privacy of its input data and internal processes. Privacy-preserving multi-agent planning allows the definition of factors of the global planning problem private to the respective entities (i.e. agents).

In such privacy-preserving planning systems (Nissim and Brafman 2012; 2014; Maliah, Shani, and Stern 2014; Torreño, Onaindia, and Sapena 2014; Tožička, Jakubův, and Komenda 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 from its view on the global problem, its projection. Such projection also contains view of other agent's public operators, which allows for heuristic estimate of the entire problem, but such estimate may be significantly misguided as shown in (Štolba, Fišer, and Komenda 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 still preserving privacy. A number of inadmissible heuristics has been treated this way such as the FF heuristic (Štolba and Komenda 2014), a DTG-based heuristic (Torreño, Onaindia, and Sapena 2014) and a landmark-based heuristic (Maliah, Shani, and Stern 2014). The admissible LM-Cut heuristic (Helmert and Domshlak 2009) is computed in a distributed way in (Štolba, Fišer, and Komenda 2015) and in (Maliah, Shani, and Stern 2015), the authors distribute an admissible pattern database heuristic, but without any proofs of admissibility of the distributed version. MAD-A* (Nissim and Brafman 2012) is the only optimal privacy-preserving multi-agent planning algorithm we are aware of.

In techniques developed so far, the distributed computation of heuristic estimate requires 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. In (Nissim and Brafman 2014), the authors propose 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 take the idea a step further, so that estimates from all agents can be added together without violating admissibility and with no additional communication needed.

Formalism

Similarly as MA-STRIPS (Brafman and Domshlak 2008) is an extension of STRIPS (Fikes and Nilsson 1971) towards privacy and multi-agent planning, we now present MA-MPT as a multi-agent extension of the Multi-Valued Planning Task (Helmert 2006) or SAS+ (Bäckström 1992). For n agents, the MA-MPT problem consists of a set of n agent MPTs. The planning task for an agent $\alpha_i \in \mathcal{A}$ is a tuple

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

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

Each V in the finite set of variables $\mathcal{V}^i$ has a finite domain of values $\mathsf{dom}(V)$. 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 *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 *global state* is a state over $\mathcal{V}^{\mathsf{G}} = \bigcup_{i\in 1..n} \mathcal{V}^i$. A global state s 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 *α_i-projected state* s^{α_i} 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). Formally, a (partial) assignment p is *consistent* with a (partial) assignment p' iff $p[V] = p'[V]$ for all $V \in \mathsf{vars}(p)$.

An *operator* o from the finite set $\mathcal{O}^i$ has a precondition $\mathsf{pre}(o)$ and effect $\mathsf{eff}(o)$ which are both partial variable assignments. In the case of private operators $o \in \mathcal{O}^{\mathsf{priv}_i}$, the assignment is 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 other agents. A function $\mathsf{cost}^i : \mathcal{O}^i \mapsto \mathbb{R}_0^+$ assigns a *cost* to each operator of agent α_i. An operator o is applicable in state s if $\mathsf{pre}(o)$ is consistent with s. Application of operator o in a state s results in 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. We denote the application of o in s as $s' = o\,[[s]]$. 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 α_i-projected (partial) states.

We define a *global problem* 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 *α_i-projected problem* is the problem of an agent α_i with α_i-projections of all other agents' public operators and it can be also understood as a complete view of the agent α_i on the global problem Π^{G}. Formally, for a public operator $o \in \mathcal{O}^{\mathsf{pub}_j}$ of some agent α_j, an *α_i-projected operator* o^{α_i} is o with precondition and effect restricted to the variables in $\mathcal{V}^i$, that is $\mathsf{pre}(o^{\alpha_i})$ is a partial variable assignment over $\mathcal{V}^i$ consistent with $\mathsf{pre}(o)$ ($\mathsf{eff}(o)$ treated analogously). The set of α_i-projected operators is

$$\mathcal{O}^{\alpha_i} = \mathcal{O}^i \cup \{o^{\alpha_i} | o \in \bigcup_{j\in 1\ldots n \wedge j\neq i} \mathcal{O}^{\mathsf{pub}_j}\}$$

and an *α_i-projected problem* is

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

where cost^{α_i} is $\mathsf{cost}^{\mathsf{G}}$ restricted to operators in $\mathcal{O}^{\alpha_i}$.

Privacy

What exactly are the agents in privacy-preserving multi-agent planning trying to hide and why? Let us consider a following example. A company has a secret recipe to a well known beverage. In order to work effectively, it wants to optimize its process of logistics and its use of subcontractors. In this example, parts of the recipe can be represented as actions either private (the most secret parts) or public (the "interface" with other companies). Other agents have the actions for logistic transportation and providing ingredients. Here, the beverage agent wants to hide all its private actions and the private parts of the public actions, namely the "signature" of the actions, that is their preconditions and effects regardless of renaming (formally, an isomorphic model).

We borrow the formal treatment of privacy-preserving planning from (Nissim and Brafman 2014). For each agent $\alpha_i \in \mathcal{A}$, the private part of its problem Π^i is

(i) the set of private variables $\mathcal{V}^{\mathsf{priv}_i}$, i.e. the number of variables and the sizes and values of their respective domains,

(ii) the set of private operators $\mathcal{O}^{\mathsf{priv}_i}$, i.e. the number of operators, the number and value of facts in $\mathsf{pre}(o)$ and $\mathsf{eff}(o)$, the value of $\mathsf{cost}^i(o)$ and

(iii) the private parts of the public operators in $\mathcal{O}^{\mathsf{pub}_i}$, i.e. the number and value of private facts in $\mathsf{pre}(o)$ and $\mathsf{eff}(o)$.

The public parts of operators in $\mathcal{O}^{\mathsf{pub}_i}$ can be shared in the form of projections.

As often assumed in cryptography, we are interested in the properties of each algorithm performed by a "honest but curious" agent. Such agent may use its computational resources to analyze the communicated data and infer as much information as possible, but the agent is not altering the protocol in order to exploit it and obtain more information. In case of the distributed search this would mean e.g., sending all possible states to other agents even though they are not reachable in order to obtain all heuristic estimates.

According to (Nissim and Brafman 2014), a *weak privacy-preserving* algorithm is such a distributed algorithm that does not directly communicate any private part of the agents' problems. By directly communicate, we mean that the information is communicated in such a way that any other agents can understand it (i.e. not encrypted).

A *strong privacy-preserving* algorithm is such a distributed algorithm that no agent α_i can deduce an isomorphic (that is differing only in renaming) model of a private variable, a private operator and its cost or private precondition and effect of a public operator belonging to some other agent α_j, beyond what can be deduced from the projected problem and the solution.

The MAD-A* multi-agent heuristic search algorithm is at least weak privacy preserving, as it does not communicate any private information. In its original form, it may be possible to deduce equality of a subset of private states and macro transitions between public states. In (Brafman 2015), the MAD-A* algorithm was modified not to send some redundant states (that is states differing only in the sending agent's private part). This modification improves the privacy guarantees for the logistics domain, such that under

additional assumptions (namely that every private location is reachable from every other private location) the algorithm is strong privacy-preserving.

Potential Heuristics

Potential heuristics are a family of admissible heuristics introduced in (Pommerening et al. 2015). Here we describe the original centralized version. A potential heuristic (denoted as h_{pot}) associates numerical potential with each fact. The potential heuristic for a state s is simply a sum of potentials of the facts in s, formally:

$$h_{\mathsf{pot}}(s) = \sum_{V \in \mathcal{V}} \mathsf{pot}(\langle V, s[V] \rangle)$$

where $\mathcal{V}$ is a set of variables and $\mathsf{pot}(\langle V, s[V] \rangle) \in \mathbb{R}$ is a potential for the fact representing the assignment for V in s.

The potentials can be determined as a solution to a linear program (LP). In this work, we use a formulation 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. Another option is to use the set of all syntactic states[1] (S), as described in (Seipp, Pommerening, and Helmert 2015), that is for all facts the coefficient associated with the potential variable of fact $\langle V, v \rangle$ is $1/|\mathsf{dom}(V)|$.

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

$$\mathsf{pot}(\langle V, v \rangle) \leq \mathsf{maxpot}_V \tag{1}$$

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

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

restricting the heuristic of any goal state to be less or equal to 0. The final set of constraints ensures consistency[2] of the heuristic, which together with the goal-awareness implies admissibility. For each operator o in a set of operators $\mathcal{O}$ we add the following constraint

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

A solution of the LP yields the values for potentials which are then used in the heuristic computation.

[1] Such LP formulation may be unbounded. A common solution we adopt is to use an upper bound for each LP variable.

[2] A consistent heuristic is such h that for each two states s, s' and all operators s.t. $s' = o[[s]]$ holds $h(s) \leq h(s') + \mathsf{cost}(o)$.

Potential Heuristics for Multi-Agent Planning

Let us first examine the global potential heuristic $h^{\mathsf{G}}_{\mathsf{pot}}$ computed on Π^{G}. For now, assume we already have the potentials for the global problem. From now on, any function (e.g., a heuristic) computed on a state s by an agent α_i is in fact computed on the α_i-projected state s^{α_i} and we omit the superscript for brevity, except for situation where it is necessary to distinguish multiple agents. For a state s, the heuristic is

$$h^{\mathsf{G}}_{\mathsf{pot}}(s) = \sum_{V \in \mathcal{V}^{\mathsf{G}}} \mathsf{pot}(\langle V, s[V] \rangle)$$

which can be rewritten as

$$h^{\mathsf{G}}_{\mathsf{pot}}(s) = \sum_{V \in \mathcal{V}^{\mathsf{pub}}} \mathsf{pot}(\langle V, s[V] \rangle) + \sum_{\alpha_i \in \mathcal{A}} \sum_{V \in \mathcal{V}^{\mathsf{priv}_i}} \mathsf{pot}(\langle V, s[V] \rangle)$$

which is the sum of potentials of public facts plus the sum of potentials of private facts of each agent. Further on, we will denote

$$h^{\mathsf{pub}}_{\mathsf{pot}}(s) = \sum_{V \in \mathcal{V}^{\mathsf{pub}}} \mathsf{pot}(\langle V, s[V] \rangle)$$

and

$$h^{\mathsf{priv}_i}_{\mathsf{pot}}(s) = \sum_{V \in \mathcal{V}^{\mathsf{priv}_i}} \mathsf{pot}(\langle V, s[V] \rangle)$$

thus the global heuristic can be rewritten as

$$h^{\mathsf{G}}_{\mathsf{pot}}(s) = h^{\mathsf{pub}}_{\mathsf{pot}}(s) + \sum_{\alpha_i \in \mathcal{A}} h^{\mathsf{priv}_i}_{\mathsf{pot}}(s). \tag{2}$$

Now we formally define the desired properties of a multi-agent heuristic function and show that the properties hold for the potential heuristic.

Definition 1. A global heuristic h estimating the global problem Π^{G} is *$\mathcal{A}$-additive* iff for any agent $\alpha_k \in \mathcal{A}$ it can be represented as $h(s) = h^{\mathsf{pub}}(s^{\alpha_k}) + \sum_{\alpha_i \in \mathcal{A}} h^i(s^{\alpha_i})$, where h^{pub} is a heuristic computed on the α_k-projected problem Π^{α_k} and h^i is a heuristic computed on the α_i-projected problem Π^{α_i}.

Each part of an $\mathcal{A}$-additive heuristic can be computed by each respective agent separately an then added together.

Definition 2. An $\mathcal{A}$-additive global heuristic $h(s) = h^{\mathsf{pub}}(s^{\alpha_k}) + \sum_{\alpha_i \in \mathcal{A}} h^i(s^{\alpha_i})$ is *$\mathcal{A}$-agnostic* iff for each two global states s and s', s.t. $s' = o[[s]]$, where $o \in \mathcal{O}^i$ for some agent α_i, holds $h^j(s^{\alpha_j}) = h^j(s'^{\alpha_j})$ for all $j \neq i$.

In an $\mathcal{A}$-agnostic heuristic, no agent can influence the private parts of other agents.

Theorem 3. *The global potential heuristic $h^{\mathsf{G}}_{\mathsf{pot}}(s) = h^{\mathsf{pub}}_{\mathsf{pot}}(s) + \sum_{\alpha_i \in \mathcal{A}} h^{\mathsf{priv}_i}_{\mathsf{pot}}(s)$ is admissible, $\mathcal{A}$-additive and $\mathcal{A}$-agnostic.*

Proof. Admissibility follows directly from the construction of the LP which is equal to the LP in the centralized case. The $\mathcal{A}$-additivity of the potential heuristic for any agent α_k follows from setting $h^{\mathsf{pub}}(s^{\alpha_k}) = h^{\mathsf{pub}}_{\mathsf{pot}}(s^{\alpha_k})$ and $h^i(s^{\alpha_i}) = h^{\mathsf{priv}_i}_{\mathsf{pot}}(s^{\alpha_i})$ for all $\alpha_i \in \mathcal{A}$.

Let $i \neq j$, the $\mathcal{A}$-agnostic property holds for $h^{\mathsf{G}}_{\mathsf{pot}}(s)$, because for each $o \in \mathcal{O}^i$, $\mathsf{eff}(o) \cap \mathcal{V}^{\mathsf{priv}_j} = \emptyset$. The part of

the state s private to agent α_j, that is partial assignment $p = s \cap \mathcal{V}^{\mathsf{priv}_j}$, is equal to the part of the successor state $s' = o[[s]]$ private to agent α_j (that is partial assignment $p' = s' \cap \mathcal{V}^{\mathsf{priv}_j}$). As both $h^{\mathsf{priv}_j}_{\mathsf{pot}}(s)$ and $h^{\mathsf{priv}_j}_{\mathsf{pot}}(s')$ are computed only on the respective parts of the states private to agent α_j, the heuristic estimates are equal. □

By application of Theorem 3, the global heuristic estimate of a successor state s' after application of an operator $o \in \mathcal{O}^i$ can be effectively computed by the agent α_i by computing the public part of the heuristic estimate, the part private to agent α_i and adding the private parts of other agents from the predecessor state s. The multi-agent distributed A* search using this principle is outlined in Figure 1.

Let us consider a concrete example with two agents α_1, α_2, where α_1 has one variable $V_1 \in \{d_1, d'_1\}$, α_2 one variable $V_2 \in \{d_2, d'_2\}$ and there is one public variable $V_{\mathsf{pub}} \in \{d_{\mathsf{pub}}, d'_{\mathsf{pub}}\}$. Assume the computed potentials are the following: $\mathsf{pot}(\langle V_1, d_1\rangle) = 1$, $\mathsf{pot}(\langle V_1, d'_1\rangle) = -2$, $\mathsf{pot}(\langle V_2, d_2\rangle) = 2$, $\mathsf{pot}(\langle V_2, d'_2\rangle) = 0$, $\mathsf{pot}(\langle V_{\mathsf{pub}}, d_{\mathsf{pub}}\rangle) = 3$, $\mathsf{pot}(\left\langle V_{\mathsf{pub}}, d'_{\mathsf{pub}}\right\rangle) = -1$. In the initial state s_I holds $s_I[V_1] = d_1$, $s_I[V_2] = d_2$ and $s_I[V_{\mathsf{pub}}] = d_{\mathsf{pub}}$, then $h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_I) = 1$, $h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_I) = 2$ and $h^{\mathsf{pub}}_{\mathsf{pot}}(s_I) = 3$, thus $h_{\mathsf{pot}}(s_I) = h^{\mathsf{pub}}_{\mathsf{pot}}(s_I) + h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_I) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_I) = 6$. If agent α_1 applies an action $a_1 \in \mathcal{O}^{\mathsf{pub}_1}$ which changes both V_1, V_{pub}, thus $s_1[V_1] = d'_1$ and $s_1[V_{\mathsf{pub}}] = d'_{\mathsf{pub}}$, the heuristic value of the resulting state s_1 computed by α_1 is $h_{\mathsf{pot}}(s_1) = h^{\mathsf{pub}}_{\mathsf{pot}}(s_1) + h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_I) = -1$. The state s_1 is then sent to agent α_2 together with the value of $h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1) = -2$. When agent α_2 applies action $a_2 \in \mathcal{O}^{\mathsf{priv}_2}$ which modifies only V_2 so that $s_2[V_2] = d'_2$, α_2 can compute $h_{\mathsf{pot}}(s_2) = h^{\mathsf{pub}}_{\mathsf{pot}}(s_2) + h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_2) = -3$ using the value of $h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1)$ received from α_1.

Before discussing how the $\mathcal{A}$-agnostic principle can be utilized in a multi-agent heuristic search, we first analyze privacy properties of the computation of potentials.

Projections

The simplest idea is to let each agent compute the LP on the projected problem and to use the resulting potentials in the global heuristic. Unfortunately, this approach does not result in an admissible heuristic. One of the reasons lies in the goal-awareness constraint, which in the global LP contains either potential or maximum potential LP-variable for each variable in $\mathcal{V}^{\mathsf{G}}$. In the projected problem, the private variables of other agents are missing. This allows the remaining variables to have higher values and results in higher potentials making the sum of private parts inadmissible.

An admissible variant is to take maximum of the projections. The α_i-projected heuristic can be rewritten as

$$\begin{aligned} h^{\alpha_i}_{\mathsf{pot}}(s) &= h^{\mathsf{pub}_i}_{\mathsf{pot}}(s) + h^{\mathsf{priv}_i}_{\mathsf{pot}}(s) \\ &= \sum_{V \in \mathcal{V}^{\mathsf{pub}}} \mathsf{pot}^i(\langle V, s[V]\rangle) + \sum_{V \in \mathcal{V}^{\mathsf{priv}_i}} \mathsf{pot}^i(\langle V, s[V]\rangle) \end{aligned}$$

where $\mathsf{pot}^i(\langle V, v\rangle)$ is the potential for fact $\langle V, v\rangle$ computed on the α_i-projected problem. Although $\mathcal{V}^{\mathsf{pub}}$ is the same for

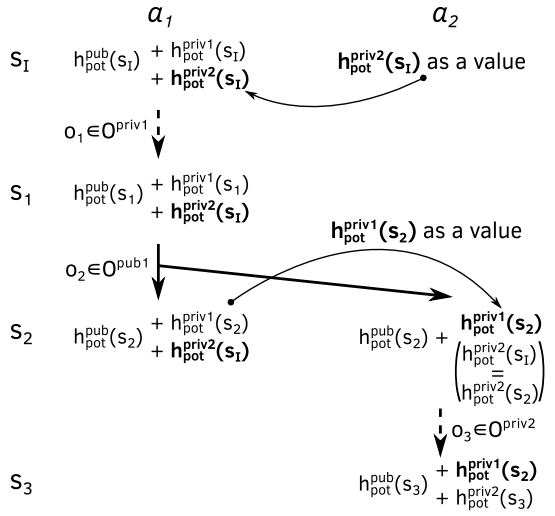

Figure 1: Sequence of $h^{\mathsf{G}}_{\mathsf{pot}}$ computations. The emphasized heuristics are not computed, but used as a heuristic value from the other agent.

all agents the potentials in $h^{\alpha_i}_{\mathsf{pot}}$ for variables in $\mathcal{V}^{\mathsf{pub}}$ may differ as they are computed using different LPs. Nevertheless, the potentials $\mathsf{pot}^i(\langle V, v\rangle)$ for public variables $V \in \mathcal{V}^{\mathsf{pub}}$ computed by agent α_i can be communicated[3] to all other agents α_j and thus the public part of the α_i-projected heuristic can be computed by each agent α_j on its projected problem Π^{α_j}. Thanks to the $\mathcal{A}$-agnostic property of the potential heuristic shown by Theorem 3 (which trivially holds also for projections), the private parts of the α_i-projected heuristic are not changed by actions of other agents. The private part can be computed by the respective agent α_i and sent along with each state s.

This means, that unlike a general projected heuristic, each agent α_i can compute the $h^{\alpha_j}_{\mathsf{pot}}$ projected heuristics of all agents and take the maximum, while still preserving privacy. We denote the resulting heuristic as

$$h^{\mathsf{maxproj}}_{\mathsf{pot}}(s) = \max_{\alpha_i \in \mathcal{A}} h^{\alpha_i}_{\mathsf{pot}}(s^{\alpha_i}).$$

To compute the $h^{\mathsf{maxproj}}_{\mathsf{pot}}(s)$ heuristic, the agent computes all public parts, its own private part, sums the corresponding public and private parts and takes the maximum, as shown in Figure 2. Obviously, $h^{\mathsf{maxproj}}_{\mathsf{pot}}(s)$ is always at least as informed as $h^{\alpha_i}_{\mathsf{pot}}(s)$, but never more informed than $h^{\mathsf{G}}_{\mathsf{pot}}(s)$.

Let us consider the example from previous section again. Now each agent has its potentials (including the public ones) computed independently, thus for agent α_1 we can have $\mathsf{pot}^1(\langle V_{\mathsf{pub}}, d_{\mathsf{pub}}\rangle) = 4$, $\mathsf{pot}^1(\left\langle V_{\mathsf{pub}}, d'_{\mathsf{pub}}\right\rangle) = 0$ and $\mathsf{pot}^1(\langle V_1, d_1\rangle) = 1$, $\mathsf{pot}^1(\langle V_1, d'_1\rangle) = -2$ and for α_2 $\mathsf{pot}^2(\langle V_{\mathsf{pub}}, d_{\mathsf{pub}}\rangle) = 3$, $\mathsf{pot}^2(\left\langle V_{\mathsf{pub}}, d'_{\mathsf{pub}}\right\rangle) = 1$ and

[3] Such potentials are influenced by constraints respective to public actions and by the goal constraint. Although the latter may possibly leak some private information, we consider sharing potentials of public variables safe.

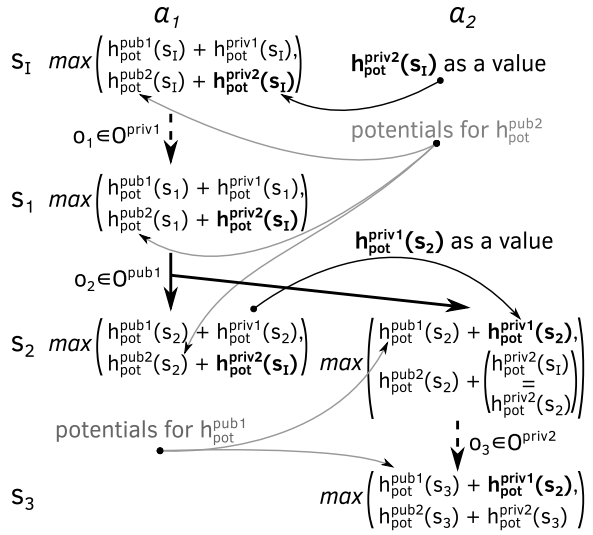

Figure 2: Sequence of $h^{\mathsf{maxproj}}_{\mathsf{pot}}$ computations. The emphasized heuristics are not computed, but used as a heuristic value from the other agent. The gray potentials are communicated from the other agent and computed as $h^{\mathsf{pub}_i}_{\mathsf{pot}}(s)$.

$\mathsf{pot}^2(\langle V_2, d_2\rangle) = 2$, $\mathsf{pot}^2(\langle V_2, d_2'\rangle) = 1$. The potentials may be higher as the LPs of the projected problems are less constrained. By sharing the public potentials, both agents can compute $h^{\mathsf{maxproj}}_{\mathsf{pot}}(s_I) = \mathsf{max}(h^{\mathsf{pub}_1}_{\mathsf{pot}}(s_I) + h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_I), h^{\mathsf{pub}_2}_{\mathsf{pot}}(s_I) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_I)) = \mathsf{max}(4+1, 3+2) = 5$. After the application of a_1 by α_1, sending the resulting state s_1 to α_2 together with the value of $h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1)$ and application of a_2 by α_2, α_2 can compute $h^{\mathsf{maxproj}}_{\mathsf{pot}}(s_2) = \mathsf{max}(h^{\mathsf{pub}_1}_{\mathsf{pot}}(s_2) + h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1), h^{\mathsf{pub}_2}_{\mathsf{pot}}(s_2) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_2)) = \mathsf{max}(0-2, 1+1) = 2$.

Plain Global LP

A baseline approach to the global LP computation is to compute it plainly as it is. The principle of the computation is simple, one agent is selected to be the master, all other agents send their private parts of the LP (that is optimization function, LP-variables and constraints) to the master. The master then solves the complete LP and sends the computed values of the LP-variables back to their respective owners.

Even in such a simple setting some privacy is preserved. This is due to the fact that the LP does not reflect preconditions of actions on variables which are not also in the effect. This means, that it is not possible to reconstruct the complete isomorphic model of an operator o, if o has some variable in precondition which is not present in effect. Nevertheless, the LP represents a planning task in the transition normal form (see (Pommerening and Helmert 2015)), which is equivalent to the original task in that the plans for the original and the transition normal form task differ only in zero cost operators (so called forgetting operators).

Decomposed Global LP

One approach to improving the privacy is to use a decomposition algorithm, such as Dantzig-Wolfe decomposition, in a similar way as proposed in (Holmgren, Persson, and Davidsson 2009). The principle of the Dantzig-Wolfe decomposition is that the problem is decomposed into a *master* part where many variables have non-zero coefficients and independent sub-matrices. For the sub-matrices hold that if a variable has a non-zero coefficient in one sub-matrix, it has zero coefficient in all other sub-matrices. If such decomposition is possible, it is used to compute the solution by iteratively generating columns for the master problem based on the sub-problem solutions.

In the case of the potential heuristic LP, the constraints for private actions, which contain only private variables would be in the sub-problem factors, whereas all other constraints would be in the master problem. In comparison to the Plain Global LP, the decomposition hides the private operators and their costs, but does not provide any formal guarantees on privacy of the computation process (the original aim of the decomposition algorithm is to improve the efficiency).

Securely Computed Global LP

Another approach is the use of a privacy-preserving transformation of the whole LP, which is often used in secure multi-party computation. Representative examples of such transformation were published in (Mangasarian 2011) and (Dreier and Kerschbaum 2011).

In (Mangasarian 2011), the transformation is applicable only on vertically partitioned data, that is data partitioned based on the variables. This means that each agent owns a disjoint subset of the LP variables and the respective parts of constraints containing them, that is, each constraint either falls completely into one partition, or is partitioned according to the variables (may span over multiple partitions). The potential heuristic LP can be partitioned in $n+1$ partitions, where the i-th partition comprises of the pot and maxpot LP variables for $V \in \mathcal{V}^{\mathsf{priv}_i}$ and the *public* $(n+1)$-th partition contains the pot and maxpot LP variables for $V \in \mathcal{V}^{\mathsf{pub}}$. The public partition, which may span over constraints of multiple agents, does not have to be encrypted and thus can be treated separately. Thus, each agent knows complete constraints for its own actions, public part of constraints for public actions and its private and public part of the goal-awareness constraint. The secure LP computation according to (Mangasarian 2011) proceeds as follows.

Let $\mathsf{max}\, c^T x$ be the optimization function and $Ax \leq b$, $A \in \mathbb{R}^{l \times m}$ the global set of constraints, which means that the global problem consists of m LP-variables and l constraints. The whole computation proceeds as follows:

1. All agents agree on some $k \geq m$. A master agent α_j which will compute the LP is selected.
2. Each agent α_i s.t. $i \neq j$ generates a random matrix $B_i \in \mathbb{R}^{k \times m_i}$, where m_i is the number of LP-variables private to agent α_i and m_{pub} is the number of public LP-variables. We define $B = [B_1 \dots B_n] \in \mathbb{R}^{k \times m}$, where B_j is a unit matrix $k \times (m_{\mathsf{pub}} + m_j)$, as α_j does not have to encrypt its part of the LP.

3. Each agent α_i sends matrix product $A_i B_i^T$ and cost coefficient product $B_i c_i$ to agent α_j, where A_i and c_i are the parts of the global LP problem private to agent α_i.
4. The linear program maximize $c^T B^T u$ subject to $AB^T = A_1 B_1^T + ... + A_n B_n^T \leq b$ is computed by agent α_j and the result vector u is sent to all other agents.
5. Each agent α_i reconstructs the solution as $x_i = B_i^T u$.

The LP for the potential heuristic differs in two features. First, there is a public part, which does not have to be encrypted. Second, some of the constraints are private-only and other agents are not aware of them. Therefore, in the Step 1 above, the agents inform the master agent about the number of constraints in the form of $k_i \geq m_i$ and k is chosen subsequently as $k = \sum_{\alpha_i \in \mathcal{A}} k_i$. This allows the agents to hide the real number of LP variables and thus also the number of variables in the private part of the planning problem (the constant k_i gives an upper bound). In Step 3 in addition to the encrypted private part, the agents send to the master also the unencrypted public part and the part of vector b respective to the private constraints (this vector encodes the costs of private actions), which are combined to form the public part of the LP and the cost vector.

The secure LP computation based on (Mangasarian 2011) with the described modifications reveals only the cost of private operators and an upper bound on their number (more constraints than the number of private actions can be sent), but without the rest of their isomorphic image. The number of variables is hidden by the k value and any information about private preconditions and effects of operators is hidden by the random matrix transformation.

More privacy-preservation can possibly be achieved by the combination of (Mangasarian 2011) and the Dantzig-Wolfe decomposition. Recall, that the decomposition is able to hide the private operator constraints and thus reduce the disclosure of the private operator costs. Nonetheless, there are no formal guarantees on the security of the process of the LP computation using the decomposition.

In (Dreier and Kerschbaum 2011) the authors generalize the transformation to arbitrarily partitioned problems and provide formal security analysis. In their approach, the agents follow a similar in idea, but rather complex protocol the description of which is out of scope of this paper. There is no information openly shared as in the previous case. There is also a low and quantifiable probability[4] of an attacker succeeding in revealing any part of the original LP, although having only inequality constraints as in our case increases the chances. Similar probability can be expected for (Mangasarian 2011), although it has not been provided in the literature.

MA Search with $\mathcal{A}$-Agnostic Heuristics

The principle of our multi-agent heuristic search is based on the MAD-A* algorithm (Multi-Agent Distributed A*) (Nissim and Brafman 2012). The algorithm is a simple extension of classical A*. The agents search in parallel, possibly

[4]The authors provide an example with 180 constraints and 282 variables, where the probabilities are below 10^{-220}.

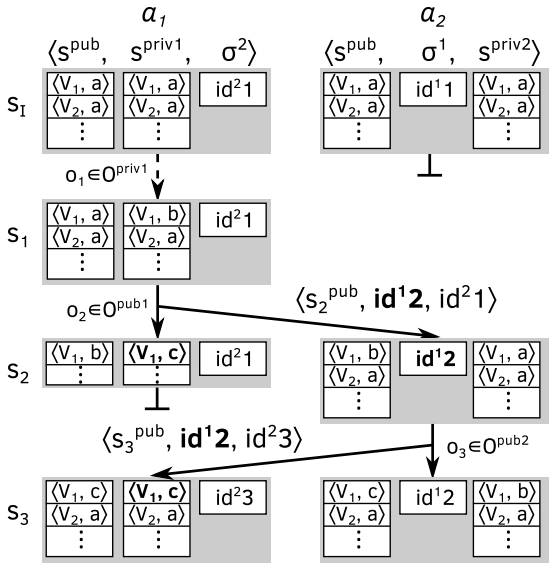

Figure 3: Search using separate public and private parts. Preserved private information $s_2^{\mathsf{priv}_1}[V_1] = \mathsf{c}$ of agent α_1 in form of identifier $\sigma_2^2 = \mathsf{id}^1 2$ used by α_2 is emphasized.

in a distributed setting (i.e. communicating over a network). Each agent $\alpha_i \in \mathcal{A}$ searches using its operators from $\mathcal{O}^i$ and if a state s is expanded using a public operator $o \in \mathcal{O}^{\mathsf{pub}_i}$, the resulting state s' is sent to other agents. 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. Each state sent by α_i is also accompanied with its α_i-projected heuristic estimate and when received, the receiving agent α_j computes α_j-projected heuristic estimate of the received state s' and takes $h(s) = \mathsf{max}(h^{\alpha_i}(s), h^{\alpha_j}(s))$.

In MA-MPT, each agent α_i can work only with its set of variables $\mathcal{V}^i$. In order to use the MAD-A* search on the MA-MPT formalism, each search state has all variables private to other agents $\alpha_{j \neq i}$ replaced by a unique identifier σ^j. This identifier refers to the last state on the search path modified by agent α_j. No other agent can reconstruct the private part from the identifier.

The search process is illustrated in Figure 3. When an agent receives a state from another agent, it uses this identifier to retrieve the proper private part. Formally, agent α_i internally represents state s as a tuple $\langle s^{\mathsf{pub}}, \sigma^1, ..., s^{\mathsf{priv}_i}, ..., \sigma^n \rangle$, where s^{pub} is the public part of the state (i.e. assignment to variables in $\mathcal{V}^{\mathsf{pub}}$), s^{priv_i} is the private part of α_i (i.e. assignment to variables in $\mathcal{V}^{\mathsf{priv}_i}$) and $\sigma^1, ..., \sigma^{i-1}, \sigma^{i+1}, ...\sigma^n$ represents the private parts of other agents. When sending a state, the private part is replaced by the respective σ^i, when received by α_j, the σ^j is replaced by s^{priv_j} from the state determined by σ^j.

In addition to the method shown in Figure 1, thanks to the $\mathcal{A}$-agnostic property of the potential heuristic, the sum of private parts of agents other than α_i can be expressed as

$h^{\mathsf{G}}_{\mathsf{pot}}(s) - h^{\mathsf{pub}}_{\mathsf{pot}}(s) - h^{\mathsf{priv}_i}_{\mathsf{pot}}(s)$ and the following holds:

$$h^{\mathsf{G}}_{\mathsf{pot}}(s') = h^{\mathsf{G}}_{\mathsf{pot}}(s) - h^{\mathsf{pub}}_{\mathsf{pot}}(s) - h^{\mathsf{priv}_i}_{\mathsf{pot}}(s) + h^{\mathsf{pub}}_{\mathsf{pot}}(s') + h^{\mathsf{priv}_i}_{\mathsf{pot}}(s')$$

where $s' = o[[s]]$ for some $o \in \mathcal{O}^i$. This means, that the heuristic estimate of a state s' can be easily determined from the heuristic estimate of its predecessor s. When a state is received from some other agent α_j, it is accompanied with its global heuristic estimate computed by α_j. When a state s is expanded, the heuristic estimate of its successor s' can be computed using the above equation. The sum in Equation 2 does not have to be explicitly computed, thus any privacy concerns of the sum computation are avoided.

Again, referring to the running example, when agent α_1 applies the public action $a_1$in s_I, resulting in s_1, it sends s_1 to α_2. Instead of sending the value of $h^{\mathsf{priv}_1}_{\mathsf{pot}}(s_1)$ as suggested previously, it can send only the value of $h^{\mathsf{G}}_{\mathsf{pot}}(s_1)$. After application of a_2, the agent α_2 can compute the heuristic estimate of s_2 simply by $h^{\mathsf{G}}_{\mathsf{pot}}(s_2) = h^{\mathsf{G}}_{\mathsf{pot}}(s_1) - h^{\mathsf{pub}}_{\mathsf{pot}}(s_1) - h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_1) + h^{\mathsf{pub}}_{\mathsf{pot}}(s_2) + h^{\mathsf{priv}_2}_{\mathsf{pot}}(s_2) = -1-(-1)-2+(-1)+0 = -3$, which equals the result obtained by the original computation.

Experimental Evaluation

For the experimental evaluation, we use the distributed track setup of the CoDMAP[5] (Štolba, Komenda, and Kovacs 2016) competition including all 12 benchmarks. Each agent runs on separate machine with i5-4460 3.4GHz processor and 8GB memory and has its own problem and domain input files. The agents communicate via TCP/IP on Gigabit Ethernet. Each run is limited to 30min.

In this section we compare the following approaches to the computation of the potential heuristic in the multi-agent setting (as a LP solver we use CPLEX 12.6.1):

$h^{\mathsf{proj}}_{\mathsf{pot}\text{-}s_I}$ The projected heuristic, that is each agent α_i computes the heuristic on its own α_i-projected problem Π^{α_i} (as in MAD-A*). The LP is optimized for the initial state s_I.

$h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$ The projected heuristic, the LP is optimized for all syntactic states.

$h^{\mathsf{maxproj}}_{\mathsf{pot}\text{-}S}$ The heuristic computed as a maximum of projections. The public potentials are shared. The LP is optimized for all syntactic states.

$h^{\mathsf{G}}_{\mathsf{pot}\text{-}s_I}$ The distributed global heuristic, the LP is optimized for the initial state s_I and with no encryption.

$h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$ The distributed global heuristic, the LP is optimized for all syntactic states and with no encryption.

$h^{\mathsf{G-sec}}_{\mathsf{pot}\text{-}S}$ The distributed global heuristic, the LP is optimized for all syntactic states and with encryption based on (Mangasarian 2011).

In the case of the secure computation based on (Mangasarian 2011), several additional matrix multiplications are performed, which does not pose significant computational overhead, as the LP computation itself is a minor part of the planning process. Also, the amount of communication is the same as when using the plain LP approach. We have measured the overhead of the secure LP computation with the following results. Over all problems, the plain LP computation takes on average 450ms, with maximum 4.5s, while the secure LP computation takes on average 520ms, with maximum 5s. This means that the secure LP computation is on average only $1.15\times$ slower than the plain LP computation. The absolute numbers show the impact on the 30min time limit is negligible for both variants of the LP computation.

This somewhat contrasts with the results shown in Table 1. The coverage of secure $h^{\mathsf{G-sec}}_{\mathsf{pot}\text{-}S}$ is nearly 15 problems less than the non-secure $h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$. Although the secure LP transformation guarantees to return optimal solution, it does not guarantee to return the same values for the LP variables (the values depend on the randomly generated matrices B_i), which may differ but still yield the same optimization function value. As different values of potentials give different heuristic estimates for the same states, the overall performance of the planner may also differ.

domain	$\lvert\mathcal{A}\rvert$	$h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{maxproj}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{G-sec}}_{\mathsf{pot}\text{-}S}$
blocksworld	4	4	4	**13**	6
depot	5 – 10	6	6	**7**	4
driverlog	2 – 8	**15**	14	**15**	13
elevators08	4	2	2	2	2
logistics00	3 – 7	4	6	**7**	6
rovers	4 – 10	1	1	1	1
satellites	3 – 8	1	1	1	1
sokoban	2 – 4	**13**	**13**	**13**	12
taxi	4 – 10	**20**	19	**20**	**20**
wireless	6 – 10	2	2	2	2
woodw.08	7	4	4	4	4
zenotravel	2 – 6	6	6	6	6
total		78	78	**91**	77

Table 1: The number of solved problems (out of 20 per domain) ($h^{\mathsf{proj}}_{\mathsf{pot}\text{-}s_I}$ and $h^{\mathsf{G}}_{\mathsf{pot}\text{-}s_I}$ solved 74 and 90 problems respectively).

domain	$\lvert\mathcal{A}\rvert$	$h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{proj}}_{\mathsf{LM\text{-}Cut}}$	$h^{\mathsf{G}}_{\mathsf{LM\text{-}Cut}}$
blocksworld	4	**13**	2	1
depot	5 – 10	**7**	6	2
driverlog	2 – 8	**15**	**15**	10
elevators08	4	**2**	**2**	0
logistics00	3 – 7	**7**	5	5
rovers	4 – 10	1	1	1
satellites	3 – 8	1	2	**3**
sokoban	2 – 4	**13**	**13**	4
taxi	4 – 10	**20**	**20**	14
wireless	6 – 10	2	**4**	3
woodw.08	7	**4**	**4**	5
zenotravel	2 – 6	6	6	6
total		**91**	80	54

Table 2: The number of solved problems (out of 20 per domain).

The secure computation based on (Dreier and Ker-

[5] http://agents.fel.cvut.cz/codmap

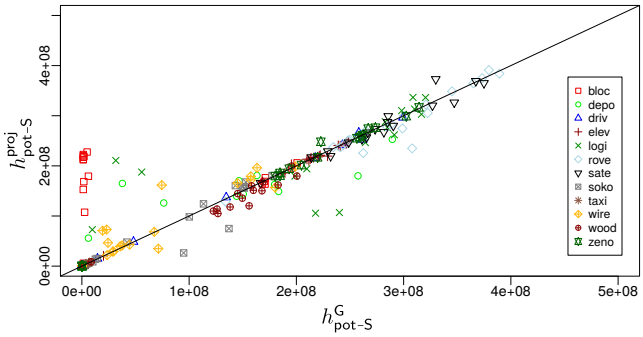

Figure 4: Ratios $h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}/h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$ of expanded states per problem.

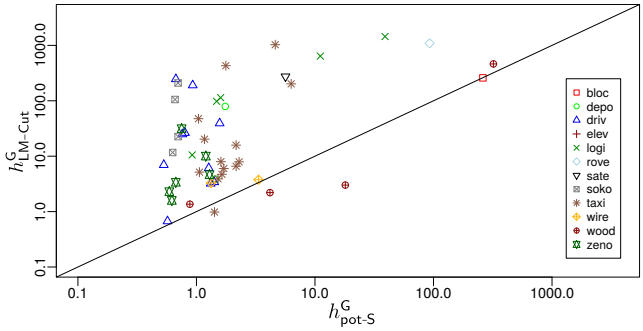

Figure 5: Time ratios $h^{\mathsf{G}}_{\mathsf{LM\text{-}Cut}}/h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$ per problem.

schbaum 2011) was not part of the experimental evaluation. Although this variant requires more matrix operations than (Mangasarian 2011), we still assume the overhead to be minimal in comparison with the MAD-A* search. Note, that vast majority of the unsolved problems are unsolved due to memory consumption and not due to reaching the time limit.

In Table 1, we present the numbers of solved problems for the variants of the potential heuristics for each competition domain ($h^{\mathsf{proj}}_{\mathsf{pot}\text{-}s_I}$ and $h^{\mathsf{G}}_{\mathsf{pot}\text{-}s_I}$ solved 74 and 90 problems in total respectively). The results show that the heuristics optimized for the set of all syntactic states (S) perform slightly better, as expected. Also the global heuristics perform better than the projected variants, as they are better informed, but does not cause any communication overheads. Even the $h^{\mathsf{maxproj}}_{\mathsf{pot}}$ variant does not bring any substantial improvement, as the global information is still missing there. The ratio of expanded states of $h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$ vs. $h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$ in Figure 4 shows that the informativeness of the global and projected heuristics is similar, except for the blocksworld, depot and logistics00 domains, which corresponds with the coverage results. In a few problems, the projected heuristic offers better guidance.

Comparison with the state of the art

Finally, we compare the MAD-A* search using the global ($h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$) potential heuristic with the state of the art. Namely, we compare it with the best performing distributed optimal multi-agent planner (Fišer, Štolba, and Komenda 2015) in CoDMAP, using a projected ($h^{\mathsf{proj}}_{\mathsf{LM\text{-}Cut}}$) and global distributed ($h^{\mathsf{G}}_{\mathsf{LM\text{-}Cut}}$) versions of the LM-Cut heuristic.

	$h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{G-sec}}_{\mathsf{pot}\text{-}S}$	$h^{\mathsf{proj}}_{\mathsf{LM\text{-}Cut}}$	$h^{\mathsf{G}}_{\mathsf{LM\text{-}Cut}}$
score	56.5	**57.9**	54.5	48.8	22.7

Table 3: IPC Agile Score.

Comparison of the number of problems solved by the planners is shown in Table 2. Whereas the performance of the projected heuristics $h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$ and $h^{\mathsf{proj}}_{\mathsf{LM\text{-}Cut}}$ is on par, the global versions indeed show the strength of $h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$, which is more informed than $h^{\mathsf{proj}}_{\mathsf{pot}\text{-}S}$ but does not incur any additional computation or communication costs. This results in a better coverage then other compared heuristics, especially the global version of LM-Cut, where the difference is over 40 problems in total.

In order to emphasize the results, let us compare results for the classical centralized versions of the heuristics in the literature. In (Pommerening et al. 2014b), the LM-Cut is reported to have coverage of 763, whereas in (Seipp, Pommerening, and Helmert 2015) the potential heuristic optimized for initial state and all syntactic states have coverage of 611 and 659 respectively (on the same experimental setting). This illustrates that although in the centralized setting, the LM-Cut heuristic performs significantly better, in the distributed setting, the properties of $h^{\mathsf{G}}_{\mathsf{pot}}$ give it a significant advantage. More advanced techniques for computing the optimization function proposed in (Seipp, Pommerening, and Helmert 2015) would probably improve the results of $h^{\mathsf{G}}_{\mathsf{pot}}$ even more.

In Table 3, the IPC Agile scores for each of the configurations are shown. The score is computed as a sum over all problem scores. For a given problem let T^* be the minimum time required by any planner to solve the problem. A configuration that solves the problem in time T gets a score of $1/(1+\log_{10}(T/T^*))$ for the problem. Search guided by any variant of the potential heuristic is faster (have higher score) than the projected LM-Cut heuristic and significantly faster than the global LM-Cut heuristic. Results for the global variants of the potential and LM-Cut heuristics are shown in Figure 5 as a per-problem ratios (restricted to problems solved by both). With a small number of exceptions, the $h^{\mathsf{G}}_{\mathsf{pot}\text{-}S}$ heuristic guided the search much faster.

Conclusion

The recently proposed class of potential heuristics proved to be a good candidate for a distributed multi-agent heuristic such that the global estimate can be computed as a sum of its local parts. This enables multi-agent search to compute better informed global estimate without any additional costs. Moreover, the heuristic can be computed in a privacy-preserving way. According to the experimental evaluation, such secure computation does not incur any significant overhead in terms of planning speed, although the difference in the resulting heuristic has impact on the performance. The newly proposed distributed heuristic outperforms the state-of-the-art distributed LM-Cut heuristic, even though it is vice versa in the centralized case. Even better performance

can be probably obtained by more elaborate techniques for selecting the optimization criterion in the LP computing of the potentials for the potential heuristic.

Acknowledgments We highly appreciate the valuable feedback from anonymous reviewers, regarding all revisions of the paper.

This research was supported by the Czech Science Foundation (grants no. 13-22125S and 15-20433Y) and by the Grant Agency of the CTU in Prague (grant no. SGS14/202/OHK3/3T/13). Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the program "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005), is greatly appreciated.

References

Bäckström, C. 1992. Equivalence and tractability results for SAS+ planning. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 126–137.

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)*, 28–35.

Brafman, R. 2015. A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the 3rd Distributed and Multiagent Planning (DMAP) Workshop of ICAPS'15*, 1–8.

Dreier, J., and Kerschbaum, F. 2011. Practical privacy-preserving multiparty linear programming based on problem transformation. In *Proceedings of IEEE 3rd International Conference on Privacy, Security, Risk and Trust (PASSAT) and IEEE 3rd Third Inernational Conference on Social Computing (SocialCom)*, 916–924.

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)*, 608–620.

Fišer, D.; Štolba, M.; and Komenda, A. 2015. MAPlan. In *Proceedings of the 1st Competition of Distributed and Multi-Agent Planners (CoDMAP)*, 8–10.

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)*, 162–169.

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

Holmgren, J.; Persson, J.; and Davidsson, P. 2009. Agent-based Dantzig-Wolfe decomposition. In *Agent and Multi-Agent Systems: Technologies and Applications*, Lecture Notes in Computer Science. Springer Berlin Heidelberg. 754–763.

Maliah, S.; Shani, G.; and Stern, R. 2014. Privacy preserving landmark detection. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, 597–602.

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*, 9–17.

Mangasarian, O. L. 2011. Privacy-preserving linear programming. *Optimization Letters* 5(1):165–172.

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)*, 1265–1266.

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

Pommerening, F., and Helmert, M. 2015. A normal form for classical planning tasks. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, 188–192.

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.; 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)*, 226–234.

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*, 3335–3341.

Seipp, J.; Pommerening, F.; and Helmert, M. 2015. New optimization functions for potential heuristics. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, 193–201.

Štolba, M., and Komenda, A. 2014. Relaxation heuristics for multiagent planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*, 298–306.

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)*, 211–219.

Štolba, M.; Komenda, A.; and Kovacs, D. L. 2016. Competition of distributed and multiagent planners (CoDMAP). In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (What's Hot Track)*.

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

Tožička, J.; Jakubův, J.; and Komenda, A. 2014. Generating multi-agent plans by distributed intersection of finite state machines. In *Proceedings of 21st European Conference on Artificial Intelligence (ECAI)*, 1111–1112.