ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230378

Gaifman Graphs in Lifted Planning

Rostislav Horčík[a;*] **and Daniel Fišer**[b]

[a]Czech Technical University in Prague, Faculty of Electrical Engineering, Prague, Czech Republic
[b]Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
ORCiD ID: Rostislav Horčík https://orcid.org/0000-0001-7967-7126,
Daniel Fišer https://orcid.org/0000-0003-2383-9477

Abstract. We introduce the metric induced by Gaifman graphs into lifted planning. We analyze what kind of information this metric carries and how it can be utilized for constructing lifted delete-free relaxation heuristics. In particular, we prove how the action dynamics influence the distances between objects. As a corollary, we derive a lower bound on the length of any plan. Finally, we apply our theoretical findings on the Gaifman graphs to improve the delete-free relaxation heuristics induced by PDDL homomorphisms.

1 Introduction

Planning on the lifted level, dealing with hard-to-ground tasks, has received attention recently. The lifted planners avoid the grounding process and search for a plan directly in the task representation given by PDDL [19]. To navigate such a lifted planner, one must introduce a lifted heuristic. Most existing methods of constructing lifted heuristics try to adapt existing methods for grounded models. For instance, by modifying the Datalog approach from the reachability analysis, the paper [1] showed how to compute $h^{\max}$, h^{add} heuristics on the lifted level. They extended their result also to the FF heuristic [3].

A distinguishing feature of lifted planning is that first-order relational structures represent the states, unlike the STRIPS formalism, where the states are just sets of structureless facts (i.e., propositions). Thus to construct lifted heuristics, it is natural to seek methods from other areas of mathematics dealing with relational structures like finite model theory [4, 17]. This paper analyzes what information Gaifman graphs can bring into lifted planning. Gaifman graphs were originally introduced in the proof of Gaifman's Theorem [7], stating that first-order logic can express only local properties of finite relational structures. Given a relational structure on a set of objects, its Gaifman graph is an undirected graph whose vertices are objects. An edge connects two objects if they occur in a ground atomic formula valid in that structure. Gaifman graphs allow us to introduce a metric on objects; for details, see [17, Chapter 4]. In the context of the IPC planning domains like transport, the metric tells us how far, for instance, a package is from a location.

To utilize the metric in lifted planning, we must understand how the metric changes between states if we apply an action. To simplify the analysis, we focus on the delete-free relaxation of PDDL tasks. Given such a task, we investigate the dynamics of the distances between objects induced by actions. We introduce a diameter of an action schema and prove that the distance between any pair of objects cannot change more than by the diameter. Consequently, we can derive a lower bound on the length of any plan.

The second part of the paper applies our theoretical results to improve delete-free relaxation heuristics introduced in [14], where it was proved that any self-map $\sigma\colon \mathcal{B} \to \mathcal{B}$ on the set of objects $\mathcal{B}$ of a PDDL task $\mathcal{P}$ is a PDDL homomorphism from $\mathcal{P}$ to a smaller PDDL task $\mathcal{P}'$, whose set of objects is $\sigma(\mathcal{B})$. Consequently, any admissible heuristic computed on $\mathcal{P}'$ induces an admissible heuristic on $\mathcal{P}$ via σ. However, they left open how to find good self-maps σ such that $\sigma(\mathcal{B})$ is sufficiently small, and the induced heuristic is informative. This paper provides a method for constructing these self-maps utilizing the metric defined by Gaifman graphs. More precisely, we look for maps σ such that $\sigma(b) = \sigma(c)$ only if b and c are close. Intuitively, we do not want to identify objects whose distance is large so that we do not create shortcuts between distant states in our state space. Our theoretical results support this intuitive idea because identifying distant objects degrades our lower bound on the plan length.

The paper is organized as follows. Section 2 recalls all necessary details on the first-order logic and PDDL planning tasks. Section 3 introduces Gaifman graphs and the metric they induce. Section 4 proves the main results, particularly the lower bound for the plan length. Section 5 shows how to utilize the theoretical results to improve the heuristics based on PDDL homomorphisms. Section 6 presents our experimental results, and Section 7 discusses the achieved results and limitations of Gaifman graphs.

2 Background

2.1 First-Order Logic

We first recall a few definitions from first-order logic; see [12, Chapter 1]. Further, we introduce our notation and conventions. Given a set S, we denote a tuple $\langle s_1, \dots, s_n\rangle$ of elements from S shortly by $\vec{s}$. The i-th component of $\vec{s}$ is denoted $s_i \in S$. For a tuple $\vec{s}$, we denote $\mathrm{Set}(\vec{s})$, the set of elements occurring in $\vec{s}$. The cartesian product of k-many copies of a set S is denoted S^k.

Given two sets B, C and a map $\sigma\colon B \to C$, we will extend σ element-wise to tuples, i.e., if $\vec{b} = \langle b_1, \dots, b_n\rangle \in B^n$ then $\sigma(\vec{b}) = \langle \sigma(b_1), \dots, \sigma(b_n)\rangle \in C^n$. In order to decrease the number of parentheses in mathematical expressions, we adopt the common convention of removing parentheses in $\sigma(\vec{b})$, i.e., writing $\sigma\vec{b}$ instead. Further, we extend σ on subsets of B. For $B' \subseteq B$, we define $\sigma(B') = \{\sigma(b) \mid b \in B'\}$.

A **first-order relational language** $\mathcal{L}$ consists of a set of **variables** $\mathcal{V} = \{v_1, v_2, \dots\}$ and a set of **predicate symbols** $\mathcal{P} = \{p_1, p_2, \dots\}$,

* Corresponding Author. Email: xhorcik@fel.cvut.cz.

each predicate symbol p_i has its arity $\mathrm{ar}(p_i)$. Even though constants are allowed in PDDL, we do not consider constants as a part of the first-order language $\mathcal{L}$ to simplify our formalisms. Nevertheless, all our results can be straightforwardly reformulated for languages with constants.

As we have no functional symbols in our first-order language $\mathcal{L}$, our **atomic formulas** (shortly **atoms**) are just expressions of the form $p(\vec{v})$ where $p \in \mathcal{P}$ is a predicate symbol and $\vec{v} = \langle v_1, \ldots, v_n \rangle$ is an n-tuple of variables for $n = \mathrm{ar}(p)$. The set of all atoms is denoted $\Phi(\mathcal{V})$.

We can define ground atoms if we have a set of objects $\mathcal{B}$. Let $\sigma\colon \mathcal{V} \to \mathcal{B}$ be a map assigning objects to variables. For each atom $p(\vec{v})$, such a map defines its corresponding **ground atom** $p(\sigma\vec{v})$. The set of all ground atoms over the set of objects $\mathcal{B}$ is denoted by $\Phi(\mathcal{B}) = \{p(\sigma\vec{v}) \mid p(\vec{v}) \in \Phi(\mathcal{V}),\ \sigma\colon \mathcal{V} \to \mathcal{B}\}$.

An $\mathcal{L}$-**structure** $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ is a set of objects $\mathcal{B}$ together with a set of ground atoms $\psi \subseteq \Phi(\mathcal{B})$. The set ψ can be understood as interpretations for predicate symbols.[1]

Given two $\mathcal{L}$-structures $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ and $\mathbf{B}' = \langle \mathcal{B}', \psi' \rangle$, we say that $\mathbf{B}'$ is a **substructure** of $\mathbf{B}$ if $\mathcal{B}' \subseteq \mathcal{B}$ and $\psi' \subseteq \psi$.

Let $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ and $\mathbf{B}' = \langle \mathcal{B}', \psi' \rangle$ be two $\mathcal{L}$-structures. A map $\sigma\colon \mathcal{B} \to \mathcal{B}'$ is called a **homomorphism** from $\mathbf{B}$ to $\mathbf{B}'$ if $p(\vec{b}) \in \psi$ implies $p(\sigma\vec{b}) \in \psi'$. Equivalently, σ is a homomorphism if $\sigma(\psi) \subseteq \psi'$ where $\sigma(\psi) = \{p(\sigma\vec{b}) \in \Phi(\mathcal{B}') \mid p(\vec{b}) \in \psi\}$. We will denote the fact that σ is a homomorphism by $\sigma\colon \mathbf{B} \to \mathbf{B}'$.

Note that if we have a set of atomic formulas $\varphi(\vec{v})$ containing variables from $\mathrm{Set}(\vec{v})$, we can view it as an $\mathcal{L}$-structure $\mathbf{S}_\varphi = \langle \mathrm{Set}(\vec{v}), \varphi(\vec{v}) \rangle$. Given an $\mathcal{L}$-structure $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ and a map $\sigma\colon \mathrm{Set}(\vec{v}) \to \mathcal{B}$, we say that $\varphi(\sigma\vec{v})$ **holds** in $\mathbf{B}$ if $\varphi(\sigma\vec{v}) \subseteq \psi$; in other words, if $\sigma\colon \mathbf{S}_\varphi \to \mathbf{B}$ is a homomorphism.

2.2 *PDDL Planning Tasks*

We consider the normalized non-numeric, non-temporal PDDL tasks without conditional effects, axioms, and negative preconditions, and with all formulas being conjunctions of atoms (represented as sets of atoms). The types are modeled as unary predicates. So for each type (i.e., a set of objects), a corresponding unary predicate is interpreted by that set of objects. We also directly split the effects of PDDL actions into add effects (positive literals) and delete effects (negative literals) in the definition below to simplify the presentation.

Similarly, as we fixed a first-order relational language $\mathcal{L}$ and then defined its $\mathcal{L}$-structures, we first define a domain language $\mathcal{D}$ and then its PDDL tasks.

Definition 1. A **domain language** $\mathcal{D} = \langle \mathcal{V}, \mathcal{P}, \mathcal{A}_S \rangle$ is a first-order relational language $\langle \mathcal{V}, \mathcal{P} \rangle$ extended with a set of action symbols $\mathcal{A}_S$. Each action symbol a has its arity $\mathrm{ar}(a)$, i.e., the number of variables it depends on. We refer to structures over the language $\langle \mathcal{V}, \mathcal{P} \rangle$ as $\mathcal{D}$-structures.

Let $a \in \mathcal{A}_S$ and $\vec{v}$ denote a tuple of pair-wise distinct variables of length $\mathrm{ar}(a)$. An **action schema** $a(\vec{v})$ is a triple $a(\vec{v}) = \langle a_{\mathrm{pre}}(\vec{v}), a_{\mathrm{add}}(\vec{v}), a_{\mathrm{del}}(\vec{v}) \rangle$ where $a_{\mathrm{pre}}(\vec{v})$, $a_{\mathrm{add}}(\vec{v})$ and $a_{\mathrm{del}}(\vec{v})$ are sets of atomic formulas built up from variables $\vec{v}$, called **preconditions**, **add effects**, and **delete effects**, respectively.

[1] The $\mathcal{L}$-structures are usually defined in logic as sets of objects endowed with relations interpreting predicate symbols from $\mathcal{L}$. Here we identify these interpretations with the corresponding set of ground atoms to be closer to the notation used in planning, i.e., understanding a state as a set of ground atoms rather than an $\mathcal{L}$-structure.

Analogously to ground atoms, we define ground actions. Given a set of objects $\mathcal{B}$, a map $\sigma\colon \mathcal{V} \to \mathcal{B}$, and an action schema $a(\vec{v})$, the corresponding **ground action** $a(\sigma\vec{v})$ is created by substituting objects $\sigma\vec{v}$ for variables $\vec{v}$.

Definition 2. Let $\mathcal{D}$ be a domain language. A **normalized PDDL task** over $\mathcal{D}$ is a tuple $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle$ where $\mathcal{B}$ is a non-empty set of objects, $\mathcal{A} = \{a(\vec{v}) \mid a \in \mathcal{A}_S\}$ a set of action schemata (one action schema for each action symbol), and $\psi_I \subseteq \Phi(\mathcal{B})$, $\psi_G \subseteq \Phi(\mathcal{B})$ are sets of ground atoms called **initial state** and **goal**, respectively. The task $\mathcal{P}$ is called **delete-free** if $a_{\mathrm{del}}(\vec{v}) = \emptyset$ for all its action schemata $a \in \mathcal{A}$.

For each PDDL task $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle$ over a domain language $\mathcal{D}$, one can construct its associated **delete-free relaxation** $\mathcal{P}^+ = \langle \mathcal{B}, \mathcal{A}^+, \psi_I, \psi_G \rangle$ over the same domain language $\mathcal{D}$ where $\mathcal{A}^+ = \{\langle a_{\mathrm{pre}}(\vec{v}), a_{\mathrm{add}}(\vec{v}), \emptyset \rangle \mid \langle a_{\mathrm{pre}}(\vec{v}), a_{\mathrm{add}}(\vec{v}), a_{\mathrm{del}}(\vec{v}) \rangle \in \mathcal{A}\}$.

A **state** in $\mathcal{P}$ is a set of ground atoms $s \subseteq \Phi(\mathcal{B})$. Note that each state can be understood as a $\mathcal{D}$-structure $\mathbf{S} = \langle \mathcal{B}, s \rangle$. Given an action schema $a(\vec{v})$ and a map $\sigma\colon \mathrm{Set}(\vec{v}) \to \mathcal{B}$, we say that the ground action $a(\sigma\vec{v})$ is **applicable** in the state s if $a_{\mathrm{pre}}(\sigma\vec{v}) \subseteq s$. In other words, $a(\sigma\vec{v})$ is applicable in s if σ is a homomorphism from $\mathbf{S}_{a_{\mathrm{pre}}(\vec{v})} = \langle \mathrm{Set}(\vec{v}), a_{\mathrm{pre}}(\vec{v}) \rangle$ to $\mathbf{S}$. The **resulting state** of applying an applicable action $a(\sigma\vec{v})$ in a state s is the state $a(\sigma\vec{v})[\![s]\!] = (s \setminus a_{\mathrm{del}}(\sigma\vec{b})) \cup a_{\mathrm{add}}(\sigma\vec{b})$.

A sequence of ground actions $\pi = \langle a_1(\vec{b}_1), \ldots, a_n(\vec{b}_n) \rangle$ is applicable in a state s_0 if there are states $s_1, \ldots, s_n$ such that $a_i(\vec{b}_i)$ is applicable in s_{i-1} and $s_i = a_i(\vec{b}_i)[\![s_{i-1}]\!]$ for $i \in \{1, \ldots, n\}$. The resulting state of this application is denoted $\pi[\![s_0]\!] = s_n$. Let s be a state. The sequence π is called an s-**plan** if π is applicable in s and $\pi[\![s]\!] \supseteq \psi_G$. In particular, if $s = \psi_I$ then π is called simply a **plan**.

In the case of optimal planning, we assume that for each action schema $a(\vec{v})$, there is a cost function c_a assigning a cost $\mathrm{c}_a(\vec{b})$ to the ground action $a(\vec{b})$. It allows us to define the cost of the plan π as $\mathrm{c}(\pi) = \sum_{i=1}^{n} \mathrm{c}_{a_i}(\vec{b}_i)$. Let s be a state. An s-plan with a minimum cost is called **optimal**.

Given a PDDL task $\mathcal{P}$ and a state s, we denote $h^*_{\mathcal{P}}(s)$ the cost of the optimal s-plan for $\mathcal{P}$. It is well-known that $h^*_{\mathcal{P}^+}(s) \leq h^*_{\mathcal{P}}(s)$ for every state s. Thus, computing an admissible heuristic in $\mathcal{P}^+$ also gives us an admissible heuristic for $\mathcal{P}$. Recall that an **admissible heuristic** for $\mathcal{P}$ is a function h assigning to s a value $h(s)$ not greater than $h^*_{\mathcal{P}}(s)$.

The following definitions will be applied only for delete-free PDDL tasks. Let $\mathbf{S} = \langle \mathcal{B}, \phi \rangle$ and $\mathbf{T} = \langle \mathcal{B}, \psi \rangle$ be $\mathcal{D}$-structures. The **level** of $\mathbf{T}$ w.r.t. $\mathbf{S}$, denoted $\mathrm{level}_{\mathbf{S}}(\mathbf{T})$, is the length $|\pi|$ of the shortest sequence of ground actions $\pi = \langle a_1(\vec{b}_1), \ldots, a_n(\vec{b}_n) \rangle$ applicable in ϕ such that $\psi \subseteq \pi[\![\phi]\!]$. If there is no such sequence, we set $\mathrm{level}_{\mathbf{S}}(\mathbf{T}) = \infty$ and call $\mathbf{T}$ **unreachable** from $\mathbf{S}$. If the level is finite, $\mathbf{T}$ is said to be **reachable** from $\mathbf{S}$. Analogously, for sets of ground atoms ψ, ϕ, we call ψ reachable from ϕ if $\langle \mathcal{B}, \psi \rangle$ is reachable from $\langle \mathcal{B}, \phi \rangle$. Further, we call a $\mathcal{D}$-structure or a set of ground atoms reachable if it is reachable from the initial state.

Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle$ be a delete-free PDDL task. As $\mathcal{P}$ has no delete effect, every reachable goal state s satisfies $\psi_I \cup \psi_G \subseteq s$. Let $\mathbf{I} = \langle \mathcal{B}, \psi_I \rangle$ and $\mathbf{G} = \langle \mathcal{B}, \psi_I \cup \psi_G \rangle$. Note that $\mathrm{level}_{\mathbf{I}}(\mathbf{G})$ is the length of the shortest plan for $\mathcal{P}$.

3 Gaifman graphs

This section introduces Gaifman graphs and the metric they induce on $\mathcal{L}$-structures. The notion of a Gaifman graph of an $\mathcal{L}$-structure comes from finite model theory; see [17, Chapter 4].

Definition 3. Let $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ be an $\mathcal{L}$-structure. Its **Gaifman graph** $G_\mathbf{B}$ is the graph $\langle \mathcal{B}, E \rangle$ whose vertices are objects, and the set of edges E (i.e., a binary symmetric relation) is defined by $\langle c, c' \rangle \in E$ iff $c = c'$ or there is a ground atom $p(\vec{b}) \in \psi$ such that $c, c' \in \mathrm{Set}(\vec{b})$.

Using the Gaifman graph $G_\mathbf{B}$ of the $\mathcal{L}$-structure $\mathbf{B}$, one can introduce a metric (distance) $\delta_\mathbf{B}$ on objects $\mathcal{B}$ by $\delta_\mathbf{B}(b, c) = 0$ if $b = c$ and otherwise $\delta_\mathbf{B}(b, c)$ is the length of the shortest path between b and c in $G_\mathbf{B}$. The distance $\delta_\mathbf{B}(b, c) = \infty$ if there is no path from b to c. Recall that the diameter of a graph is the greatest distance between any pair of vertices. We define a **diameter** $D_\mathbf{B}$ of an $\mathcal{L}$-structure $\mathbf{B}$ as the diameter of its Gaifman graph $G_\mathbf{B}$. Note that the diameter $D_\mathbf{B}$ is ∞ iff $G_\mathbf{B}$ is disconnected.

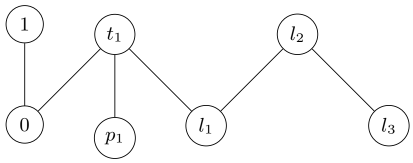


Figure 1: Gaifman graph of the $\mathcal{L}$-structure from Example 4. The loops on vertices are omitted.

Example 4. Let $\mathcal{L}$ be a predicate language from the IPC domain transport consisting of five binary predicate symbols `at`, `in`, `road`, `cap`, `pred`, and four unary predicate symbols `veh`, `pkg`, `loc` and `num`. Consider the $\mathcal{L}$-structure $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ where $\mathcal{B} = \{t_1, p_1, l_1, l_2, l_3, 0, 1\}$ and

$$\begin{aligned}\psi = \{&\mathtt{veh}(t_1), \mathtt{pkg}(p_1), \mathtt{loc}(l_1), \mathtt{loc}(l_2), \mathtt{loc}(l_3), \mathtt{num}(0), \mathtt{num}(1),\\ &\mathtt{at}(t_1, l_1), \mathtt{in}(p_1, t_1), \mathtt{cap}(t_1, 0), \mathtt{pred}(0, 1), \mathtt{road}(l_1, l_2),\\ &\mathtt{road}(l_2, l_1), \mathtt{road}(l_2, l_3), \mathtt{road}(l_3, l_2)\}.\end{aligned}$$

The Gaifman graph of $\mathbf{B}$ is depicted in Figure 1. We have $\delta_\mathbf{B}(p_1, l_3) = 4$, and the diameter $D_\mathbf{B}$ is 5. Note that unary predicates do not impact the Gaifman graph and the induced metric.

Further, consider a set of atomic formulas $\varphi(x, y, z) = \{\mathtt{veh}(x), \mathtt{at}(x, y), \mathtt{road}(y, z)\}$. We can view it as an $\mathcal{L}$-structure $\mathbf{S}_\varphi = \langle \{x, y, z\}, \varphi(x, y, z) \rangle$. Note the diameter $D_{\mathbf{S}_\varphi} = 2$. Let $\sigma\colon \{x, y, z\} \to \mathcal{B}$ be the map defined by $x \mapsto t_1$, $y \mapsto l_1$, and $z \mapsto l_2$. Then, σ is a homomorphism from $\mathbf{S}_\varphi$ to $\mathbf{B}$, and so $\varphi(t_1, l_1, l_2)$ holds in $\mathbf{B}$.

The following lemma shows how the metric on $\mathcal{L}$-structures behaves on substructures and when we apply a homomorphism.

Lemma 5. *Let $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ and $\mathbf{B}' = \langle \mathcal{B}', \psi' \rangle$ be $\mathcal{L}$-structures and $G_\mathbf{B} = \langle \mathcal{B}, E \rangle$, $G_{\mathbf{B}'} = \langle \mathcal{B}', E' \rangle$ their respective Gaifman graphs. Then the following hold:*

1. *If $\sigma\colon \mathcal{B} \to \mathcal{B}'$ is a homomorphism from $\mathbf{B}$ to $\mathbf{B}'$, then σ is a graph homomorphism from $G_\mathbf{B}$ to $G_{\mathbf{B}'}$, i.e., if $\langle c, c' \rangle \in E$ then $\langle \sigma(c), \sigma(c') \rangle \in E'$.*
2. *If $\sigma\colon \mathcal{B} \to \mathcal{B}'$ is a homomorphism from $\mathbf{B}$ to $\mathbf{B}'$, then $\delta_{\mathbf{B}'}(\sigma(b), \sigma(c)) \leq \delta_\mathbf{B}(b, c)$ for any pair of objects $b, c \in \mathcal{B}$.*
3. *If $\mathbf{B}'$ is a substructure of $\mathbf{B}$ then $\delta_\mathbf{B}(b, c) \leq \delta_{\mathbf{B}'}(b, c)$ for any pair of objects $b, c \in \mathcal{B}'$.*

Proof. (1) Suppose $\langle c, c' \rangle \in E$. If $c = c'$, then $\langle \sigma(c), \sigma(c) \rangle \in E'$ by Definition 3. If $c \neq c'$, there is a ground atom $p(\vec{b}) \in \psi$ such that $c, c' \in \mathrm{Set}(\vec{b})$. As $\sigma\colon \mathbf{B} \to \mathbf{B}'$, we have $p(\sigma\vec{b}) \in \psi'$. Consequently, $\langle \sigma(c), \sigma(c') \rangle \in E'$.

(2) Next, consider the shortest path $b = b_0, \ldots, b_n = c$ from b to c in $G_\mathbf{B}$. By (1) σ is a graph homomorphism, hence $\sigma(b) = \sigma(b_0), \ldots, \sigma(b_n) = \sigma(c)$ is a path in $G_{\mathbf{B}'}$. Consequently, $\delta_{\mathbf{B}'}(\sigma(b), \sigma(c)) \leq n = \delta_\mathbf{B}(b, c)$.

(3) It is easy to see that if $\mathbf{B}'$ is a substructure of $\mathbf{B}$, then $E' \subseteq E$. Thus each path in $G_{\mathbf{B}'}$ is a path in $G_\mathbf{B}$ as well. □

4 Main Results

In the rest of the paper, we fix a domain language $\mathcal{D}$. As we will be interested in delete-free PDDL tasks, we denote the class of all delete-free PDDL tasks over $\mathcal{D}$ as PDDL^+.

Consider a PDDL task $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+$. In this section, we analyze what information on $\mathcal{P}$ is encoded in the metric induced on $\mathcal{D}$-structures via the Gaifman graphs. We will first discuss how the pairwise distances between objects (particularly those occurring in the goal) are changing as we proceed from the initial state toward a goal state.

Let $\pi = a_1(\vec{b}_1), \ldots, a_n(\vec{b}_n)$ be a plan for $\mathcal{P}$ and $\psi_I = s_0, s_1, \ldots, s_n \supseteq \psi_G$ the corresponding sequence of states leading from the initial state to a goal state. As $\mathcal{P}$ is delete-free, we have $s_i \subseteq s_j$ for $i \leq j$. Each state s_i corresponds to a $\mathcal{D}$-structure $\mathbf{S}_i = \langle \mathcal{B}, s_i \rangle$ endowed with the metric $\delta_{\mathbf{S}_i}$ induced by its Gaifman graph $G_{\mathbf{S}_i}$.

It is easy to see that each action application cannot make the distance of any pair of objects $b, c \in \mathcal{B}$ larger because $\mathcal{P}$ has no delete effects. As $s_i \subseteq s_j$, $\mathbf{S}_i$ is a substructure of any $\mathbf{S}_j$ for $i \leq j$. Consequently, we have $\delta_{\mathbf{S}_j}(b, c) \leq \delta_{\mathbf{S}_i}(b, c)$ by Lemma 5(3). Thus the objects get closer when we successively apply the actions from π.

Next, let us see what conditions on the distances are implied by the goal ψ_G. Let $\mathbf{G} = \langle \mathcal{B}, \psi_I \cup \psi_G \rangle$ be the $\mathcal{D}$-structure defined by the goal, and $\delta_\mathbf{G}$ the corresponding metric. Note that the $\mathcal{D}$-structure $\mathbf{G}$ must be a substructure of $\mathbf{S}_n$ because $s_n \supseteq \psi_I \cup \psi_G$ is a goal state. The metric $\delta_\mathbf{G}$ expresses how far the goal objects are required to be from each other at most in a goal state. Indeed, by Lemma 5(3), we have $\delta_{\mathbf{S}_n}(b, c) \leq \delta_\mathbf{G}(b, c)$ for any $b, c \in \mathcal{B}$. Note that if $\delta_\mathbf{G}(b, c) = \infty$, the distance between b and c in a goal state might be arbitrary.

These considerations show that action applications successively shorten the distances between the goal objects until the objects are sufficiently close as specified by the goal ψ_G. We can be more precise if we quantify how much objects can get closer if we apply an action.

Definition 6. Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+$ and $a(\vec{x}) \in \mathcal{A}$ an action schema. The diameter D_a of the $\mathcal{D}$-structure $\mathbf{S}_{a_{\mathrm{pre}}(\vec{x})} = \langle \mathrm{Set}(\vec{x}), a_{\mathrm{pre}}(\vec{x}) \rangle$ is called the **diameter** of $a(\vec{x})$.

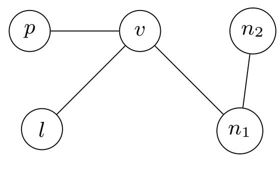


Figure 2: The Gaifman graph corresponding to the action schema `drop`.

Example 7. Consider the action schema `drop` from the IPC domain transport. Its parameters are a vehicle v, a location l, a package p, and capacity numbers n_1, n_2. The preconditions are formed by the set of atoms $\varphi = \{\mathtt{at}(v, l), \mathtt{in}(p, v), \mathtt{cap}(v, n_1), \mathtt{pred}(n_1, n_2)\}$; the unary predicates expressing that v is a vehicle, l a location, etc., are omitted as they are irrelevant to the Gaifman graph. Let $\vec{x} = \langle v, l, p, n_1, n_2 \rangle$. The Gaifman graph of the structure $\mathbf{S}_{\mathtt{drop}_{\mathrm{pre}}(\vec{x})} = \langle \mathrm{Set}(\vec{x}), \varphi \rangle$ is depicted in Figure 2. Thus the diameter of `drop` is 3.

The following results assume that the action-schema diameters are finite. If an action schema a has an infinite diameter, the analysis through distances is limited because a can arbitrarily change the distance between objects.[2] In particular, disconnected objects might become connected by the application. However, if the diameter of a is finite, the distance changes can be bound by the diameter.

Theorem 8. *Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+$, $c, c' \in \mathcal{B}$ objects, s a state, $a(\vec{x}) \in \mathcal{A}$ an action schema and $\sigma\colon \mathrm{Set}(\vec{x}) \to \mathcal{B}$ such that $a(\sigma\vec{x})$ is applicable in s. Let $s' = a(\sigma\vec{x})[\![s]\!]$ be the resulting state and $\delta_{\mathbf{S}}$ (resp. $\delta_{\mathbf{S}'}$) the metric on the $\mathcal{D}$-structure $\mathbf{S} = \langle \mathcal{B}, s \rangle$ (resp. $\mathbf{S}' = \langle \mathcal{B}, s' \rangle$). If $\delta_{\mathbf{S}'}(c, c') < \infty$, then $\delta_{\mathbf{S}}(c, c') - \delta_{\mathbf{S}'}(c, c') \leq \max\{0, D_a - 1\}$ where D_a is the diameter of $a(\vec{x})$.*

Proof. The claim clearly holds if $\delta_{\mathbf{S}}(c, c') = \delta_{\mathbf{S}'}(c, c')$. So suppose that $\delta_{\mathbf{S}}(c, c') > \delta_{\mathbf{S}'}(c, c')$. It follows that $c \neq c'$. As $a(\sigma\vec{x})$ is applicable in s, σ is a homomorphism from $\mathbf{S}_{a_{\mathrm{pre}}(\vec{x})}$ to $\mathbf{S}$. Consider the shortest path $c = c_0, \ldots, c_n = c'$ in $G_{\mathbf{S}'}$ whose length is $n = \delta_{\mathbf{S}'}(c, c')$. As the path is shorter than $\delta_{\mathbf{S}}(c, c')$, the path must include at least one edge introduced by $a_{\mathrm{add}}(\sigma\vec{x})$. Consequently, there must be the least index j and the greatest index k such that $c_j, c_k \in \mathrm{Set}(\sigma\vec{x})$ and $j < k$. Let $c_j = \sigma(x_i)$ and $c_k = \sigma(x_r)$ for some $x_i, x_r \in \mathrm{Set}(\vec{x})$. As the diameter of $\mathbf{S}_{a_{\mathrm{pre}}(\vec{x})}$ is D_a, the distance between x_i and x_r is at most D_a in $\mathbf{S}_{a_{\mathrm{pre}}(\vec{x})}$. Hence $\delta_{\mathbf{S}}(c_j, c_k) \leq D_a$ by Lemma 5(2). Consequently, we can replace the subpath from c_j to c_k in $G_{\mathbf{S}'}$ (whose length is at least 1) with a path inside $G_{\mathbf{S}}$ of length at most D_a. The resulting sequence of edges is a path in $G_{\mathbf{S}}$. Thus we have the desired $\delta_{\mathbf{S}}(c, c') \leq n - 1 + D_a = \delta_{\mathbf{S}'}(c, c') + D_a - 1$. □

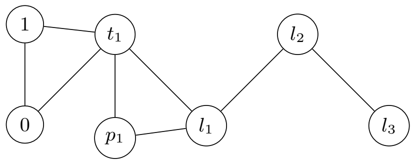


Figure 3: Gaifman graph of the $\mathcal{L}$-structure from Example 4 after the application of the delete-free variant of the action `drop`.

Example 9. Consider the structure $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ from Example 4 in the domain language of the IPC domain transport. The preconditions of the action `drop` hold for $v \mapsto t_1$, $l \mapsto l_1$, $p \mapsto p_1$, $n_1 \mapsto 0$, and $n_2 \mapsto 1$. If we apply the delete-free version of `drop` in $\mathbf{B}$, we get an $\mathcal{L}$-structure $\mathbf{B}' = \langle \mathcal{B}, \psi \cup \{\mathtt{at}(p_1, l_1), \mathtt{cap}(t_1, 1)\} \rangle$. Its Gaifman graph is shown in Figure 3. We have $\delta_{\mathbf{B}}(p_1, l_3) - \delta_{\mathbf{B}'}(p_1, l_3) = 4 - 3 = 1 \leq 2 = D_{\mathtt{drop}} - 1$.

In the following, we focus on delete-free PDDL tasks over a domain language $\mathcal{D}$ whose actions have a finite diameter and at least one action schema of a nonzero diameter.[3] A class of such PDDL tasks where the maximum diameter $D = max_{a(\vec{x}) \in \mathcal{A}} D_a > 0$ is denoted PDDL^+_D. As $D > 0$, we have $\max\{0, D_a - 1\} \leq D - 1$ for any $a(\vec{x}) \in \mathcal{A}$.

[2] We will discuss what can be done with action schemata of the infinite diameter in Section 7.

[3] The tasks having only action schemata with a zero diameter are rather trivial as its actions can have only a single parameter x and its add effects are of the form $p(x, \ldots, x)$ for a predicate symbol p.

Theorem 8 bounds the change of distances between objects if we apply a single action. The following theorem provides a bound if we apply a sequence of actions.

Theorem 10. *Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+_D$, $\mathbf{T} = \langle \mathcal{B}, \psi \rangle$ a reachable $\mathcal{D}$-structure from a $\mathcal{D}$-structure $\mathbf{S} = \langle \mathcal{B}, \phi \rangle$, and $b, c \in \mathcal{B}$ such that $\delta_{\mathbf{T}}(b, c) < \infty$. Then $\mathrm{level}_{\mathbf{S}}(\mathbf{T}) \geq (\delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{T}}(b, c))/(D - 1)$.*

Proof. We have $\mathrm{level}_{\mathbf{S}}(\mathbf{T}) = n$ for some $n \in \mathbb{N}$ by reachability. Thus there is a sequence of ground actions $\pi = a_1(\vec{b}_1), \ldots, a_n(\vec{b}_n)$ with the corresponding sequence of states $\phi = s_0, \ldots, s_n \supseteq \psi$. Let $\mathbf{S}_i = \langle \mathcal{B}, s_i \rangle$ be the $\mathcal{D}$-structure of the state s_i. Thus $\mathbf{T}$ is a substructure of $\mathbf{S}_n$. Consequently, $\delta_{\mathbf{S}_n}(b, c) \leq \delta_{\mathbf{T}}(b, c) < \infty$ by Lemma 5(3).

By Theorem 8, we have $\delta_{\mathbf{S}_{i-1}}(b, c) - \delta_{\mathbf{S}_i}(b, c) \leq D - 1$ for all $i = 0, \ldots, n$. In particular, $\delta_{\mathbf{S}_i}(b, c) < \infty$ for each i. We must prove that $\delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{T}}(b, c) \leq (D-1)n$. Applying the above facts, we get $\delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{T}}(b, c) \leq \delta_{\mathbf{S}_0}(b, c) - \delta_{\mathbf{S}_n}(b, c) = \sum_{i=1}^{n}(\delta_{\mathbf{S}_{i-1}}(b, c) - \delta_{\mathbf{S}_i}(b, c)) \leq (D - 1)n$. □

The lower bound on the plan length directly follows from Theorem 10.

Corollary 11. *Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+_D$, s a state with its structure $\mathbf{S} = \langle \mathcal{B}, s \rangle$, $\mathbf{G} = \langle \mathcal{B}, s \cup \psi_G \rangle$, and $b, c \in \mathcal{B}$ such that $\delta_{\mathbf{G}}(b, c) < \infty$. If there is an s-plan π, then $|\pi| \geq (\delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{G}}(b, c))/(D - 1)$.*

Proof. By Theorem 10 for $\mathbf{T} = \mathbf{G}$, we have $|\pi| \geq \mathrm{level}_{\mathbf{S}}(\mathbf{G}) \geq (\delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{G}}(b, c))/(D - 1)$. □

Example 12. Consider again the structure $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ from Example 4. We can understand it as a state in a PDDL task from the transport domain. Note that the maximum diameter D of action schemata in transport is 3. Let $\mathbf{G} = \langle \mathcal{B}, \psi \cup \{\mathtt{at}(p_1, l_3)\} \rangle$ be a goal. By Corollary 11, any plan starting in $\mathbf{B}$ and reaching a state containing $\mathbf{G}$ has a length at least 2 because $\mathrm{level}_{\mathbf{B}}(\mathbf{G}) \geq (\delta_{\mathbf{B}}(p_1, l_3) - \delta_{\mathbf{G}}(p_1, l_3))/(D - 1) = (4 - 1)/2 = 3/2$.

Theorem 10 also implies the following corollary on the reachability of facts.

Corollary 13. *Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+_D$, $\mathbf{S} = \langle \mathcal{B}, s \rangle$ a reachable $\mathcal{D}$-structure from $\mathbf{I} = \langle \mathcal{B}, \psi_I \rangle$. Then the following hold:*

1. *$\delta_{\mathbf{S}}(b, c) < \infty$ iff $\delta_{\mathbf{I}}(b, c) < \infty$ for all objects $b, c \in \mathcal{B}$.*
2. *Let $p(\vec{b})$ be a ground atom such that $\{p(\vec{b})\}$ is reachable. Then $\delta_{\mathbf{I}}(b, c) < \infty$ for all $b, c \in \mathrm{Set}(\vec{b})$.*

Proof. For the first item, we have $\delta_{\mathbf{S}}(b, c) \leq \delta_{\mathbf{I}}(b, c)$ by Lemma 5(3) because $\mathbf{I}$ is a substructure of $\mathbf{S}$. Thus, the right-to-left implication follows. Conversely, if $\delta_{\mathbf{S}}(b, c) < \infty$, then $\delta_{\mathbf{I}}(b, c) \leq (D - 1)\mathrm{level}_{\mathbf{I}}(\mathbf{S}) + \delta_{\mathbf{S}}(b, c)$ by Theorem 10. As $\mathrm{level}_{\mathbf{I}}(\mathbf{S}) < \infty$ by reachability, we have $\delta_{\mathbf{I}}(b, c) < \infty$ as well.

Suppose that $p(\vec{b})$ is a reachable ground atom for the second item. Thus there is a reachable state $\mathbf{S} = \langle \mathcal{B}, s \rangle$ such that $p(\vec{b}) \in s$. We have $\delta_{\mathbf{S}}(b, c) \leq 1$ by the metric definition. Thus $\delta_{\mathbf{I}}(b, c) < \infty$ by the previous item. □

5 Lifted Heuristic

In this section, we apply the results from the previous section in order to improve the delete-free relaxation heuristics based on PDDL-homomorphisms described in [14]. We first recall the relevant details.

Definition 14. Let $\mathcal{D}$ be a domain language, and $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle$, $\mathcal{P}' = \langle \mathcal{B}', \mathcal{A}', \psi'_I, \psi'_G \rangle$ PDDL tasks over $\mathcal{D}$. A map $\sigma\colon \mathcal{B} \to \mathcal{B}'$ is called a **PDDL homomorphism**, denoted by $\sigma\colon \mathcal{P} \to \mathcal{P}'$, if the following conditions are satisfied:

(P1) $\sigma(\psi_I) \subseteq \psi'_I$,
(P2) $\sigma(\psi_G) \supseteq \psi'_G$,
(P3) for each reachable state s in $\mathcal{P}$, each state s' in $\mathcal{P}'$ and each ground action $a(\vec{b})$ applicable in s if $\sigma\colon \langle \mathcal{B}, s \rangle \to \langle \mathcal{B}', s' \rangle$ is a homomorphism, then $a(\sigma\vec{b})$ is applicable in s' and $\sigma\colon \langle \mathcal{B}, t \rangle \to \langle \mathcal{B}', t' \rangle$ is a homomorphism for $t = a(\vec{b})[\![s]\!]$ and $t' = a(\sigma\vec{b})[\![s']\!]$.
(P4) for optimal planning, we further require that $c_a(\sigma\vec{b}) \le c_a(\vec{b})$ for all ground actions $a(\vec{b})$.

PDDL homomorphisms can be used to construct admissible heuristic due to the following theorem proved in [14, Corollary 6].

Theorem 15. *Let $\sigma\colon \mathcal{P} \to \mathcal{P}'$ be a PDDL homomorphism from a PDDL task $\mathcal{P}$ to a PDDL task $\mathcal{P}'$ and s a reachable state in $\mathcal{P}$. Then $h^*_{\mathcal{P}'}(\sigma(s)) \le h^*_{\mathcal{P}}(s)$.*

Consequently, if we are able to compute an admissible heuristic on $\mathcal{P}'$, we can use it to lower bound $h^*_{\mathcal{P}}(s)$. To find a suitable $\mathcal{P}'$, the paper [14] proposed the following construction. Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+$ be a delete-free PDDL task. Consider a self map $\sigma\colon \mathcal{B} \to \mathcal{B}$. We define a new delete-free PDDL task $\sigma(\mathcal{P}) = \langle \sigma(\mathcal{B}), \mathcal{A}, \sigma(\psi_I), \sigma(\psi_G) \rangle$ where the action costs in $\sigma(\mathcal{P})$ are defined as $c_a(\sigma\vec{b}) = \min\{c_a(\vec{c}) \mid \vec{c} \in \mathcal{B}^{\mathrm{ar}(a)},\ \sigma\vec{c} = \sigma\vec{b}\}$. The map σ is a PDDL homomorphism from $\mathcal{P}$ to $\sigma(\mathcal{P})$; see [14, Theorem 8].

Theorem 16. *Let $\mathcal{P} = \langle \mathcal{B}, \mathcal{A}, \psi_I, \psi_G \rangle \in \mathrm{PDDL}^+$ and $\sigma\colon \mathcal{B} \to \mathcal{B}$ a self map on $\mathcal{B}$. Then σ is a PDDL homomorphism from $\mathcal{P}$ to $\sigma(\mathcal{P})$. Thus, $h^*_{\sigma(\mathcal{P})}(\sigma(s)) \le h^*_{\mathcal{P}}(s)$ for each reachable state s in $\mathcal{P}$.*

One can use the above construction and theorem to define an admissible heuristic on any PDDL task. Let $\mathcal{P}$ be a PDDL task. First, consider its delete-free relaxation $\mathcal{P}^+$. Next, select a self map $\sigma\colon \mathcal{B} \to \mathcal{B}$ so that the image $\sigma(\mathcal{B})$ is small. Apply grounding to $\sigma(\mathcal{P}^+)$ and consider any admissible heuristic $h_{\sigma(\mathcal{P}^+)}$ for the grounded task. To compute the heuristic value for a reachable state s in the original problem $\mathcal{P}$, evaluate $h_{\sigma(\mathcal{P}^+)}(\sigma(s))$. This approach defines an admissible heuristic for $\mathcal{P}$ because $h_{\sigma(\mathcal{P}^+)}(\sigma(s)) \le h^*_{\sigma(\mathcal{P}^+)}(\sigma(s)) \le h^*_{\mathcal{P}^+}(s) \le h^*_{\mathcal{P}}(s)$ for any reachable state s in $\mathcal{P}$.

The paper [14] generated the self maps σ randomly and considered the map giving the most informative heuristic for the initial state. Instead of generating σ randomly, we provide a strategy for finding good self maps σ.

To narrow down possible self maps σ, we focus on retraction maps. A map $\sigma\colon \mathcal{B} \to \mathcal{B}$ is a **retraction** if it behaves like an identity on its image, i.e., we have $\sigma(b) = b$ for all $b \in \sigma(\mathcal{B})$. This choice is because we want to reduce the number of objects and not permute them. Further, we focus on self maps σ preserving the goal, i.e., $\sigma(\psi_G) = \psi_G$. Finally, we identify only objects of the same type, i.e., they must satisfy the same unary predicates modeling the types.

We construct the self map σ iteratively as a composition $\sigma = \sigma_1 \circ \cdots \circ \sigma_m$. Each σ_i identifies a single pair of objects and fixes the rest. More precisely, we find a suitable pair of objects c, c' not identified by previous maps $\sigma_1, \ldots, \sigma_{i-1}$ and define $\sigma_i(c') = c$ and $\sigma_i(b) = b$ for all $b \neq c'$. The number m depends on the object reduction we want to achieve because $|\sigma(\mathcal{B})| = |\mathcal{B}| - m$.

We want to choose σ so that $h^*_{\sigma(\mathcal{P}^+)}(\sigma(s))$ is close to $h^*_{\mathcal{P}^+}(s)$. Although Corollary 11 cannot be used to infer lower bounds directly on $h^*_{\mathcal{P}^+}(s)$ and $h^*_{\sigma(\mathcal{P}^+)}(\sigma(s))$, we can at least try to select σ so that the lower bound on the length of the $\sigma(s)$-plan in $\sigma(\mathcal{P}^+)$ is not much smaller than the lower bound on the length of the s-plan in $\mathcal{P}^+$. To compare these two lower bounds, we need to understand the relationship between the Gaifman metric on the structures $\mathbf{S} = \langle \mathcal{B}, s \rangle$ and $\sigma(\mathbf{S}) = \langle \sigma(\mathcal{B}), \sigma(s) \rangle$, respectively. Note that σ is a homomorphism from $\mathbf{S}$ to $\sigma(\mathbf{S})$. Let $b, c \in \sigma(\mathcal{B})$. As we assume that σ is a retraction, we have $\delta_{\sigma(\mathbf{S})}(b, c) = \delta_{\sigma(\mathbf{S})}(\sigma(b), \sigma(c)) \le \delta_{\mathbf{S}}(b, c)$ by Lemma 5(2). Thus the distances in $\sigma(\mathbf{S})$ are never larger than in $\mathbf{S}$. We can even quantify how much smaller they can be.

Lemma 17. *Let $\mathbf{B} = \langle \mathcal{B}, \psi \rangle$ be an $\mathcal{L}$-structure and $c, c' \in \mathcal{B}$ two objects such that $\delta_{\mathbf{B}}(c, c') = k$ for some $k \ge 1$. Consider the map $\sigma\colon \mathcal{B} \to \mathcal{B}$ such that $\sigma(c') = c$ and $\sigma(b) = b$ for all objects $b \neq c'$. Further, let $\sigma(\mathbf{B})$ be the $\mathcal{L}$-structure $\langle \sigma(\mathcal{B}), \sigma(\psi) \rangle$. Then for any $b_1, b_2 \in \sigma(\mathcal{B})$, $\delta_{\sigma(\mathbf{B})}(b_1, b_2) < \infty$ implies $\delta_{\mathbf{B}}(b_1, b_2) - \delta_{\sigma(\mathbf{B})}(b_1, b_2) \le k$.*

Proof. Let $G_{\mathbf{B}} = \langle \mathcal{B}, E \rangle$ and $G_{\sigma(\mathbf{B})} = \langle \sigma(\mathcal{B}), E' \rangle$ be the Gaifman graphs of $\mathbf{B}$ and $\sigma(\mathbf{B})$, respectively. As we identify only c' to c, the edges in E' not occurring in E are among

$$\{\langle b, c \rangle \mid b \in \sigma(\mathcal{B}), \langle b, c' \rangle \in E\} \cup \{\langle c, b \rangle \mid b \in \sigma(\mathcal{B}), \langle c', b \rangle \in E\}.$$

Suppose that $\delta_{\sigma(\mathbf{B})}(b_1, b_2) < \infty$ and consider the shortest path π with a sequence of objects $b_1 = c_0, \ldots, c_n = b_2$ in the Gaifman graph $G_{\sigma(\mathbf{B})}$ from b_1 to b_2. If π is also a path in $G_{\mathbf{B}}$, then we have $\delta_{\mathbf{B}}(b_1, b_2) = \delta_{\sigma(\mathbf{B})}(b_1, b_2)$. Suppose that π is not a path in $G_{\mathbf{B}}$. As we identify only c' to c, the shortest path π must contain c exactly once, i.e., $c = c_i$ for some $0 \le i \le n$. There are two possibilities. Either $\langle c_{i-1}, c \rangle \in E$ and $\langle c', c_{i+1} \rangle \in E$ or $\langle c_{i-1}, c' \rangle \in E$ and $\langle c, c_{i+1} \rangle \in E$. In both cases, we can connect c and c' by a path in $G_{\mathbf{B}}$ of length k by the assumption $\delta_{\mathbf{B}}(c, c') = k$. Thus we can construct a path in $G_{\mathbf{B}}$ connecting b_1 and b_2 whose length is $\delta_{\sigma(\mathbf{B})}(b_1, b_2) + k$. Consequently, $\delta_{\mathbf{B}}(b_1, b_2) \le \delta_{\sigma(\mathbf{B})}(b_1, b_2) + k$. □

As $\sigma = \sigma_1 \circ \cdots \circ \sigma_m$, we need to apply Lemma 17 m-times. Consequently, if each σ_i identifies a pair of objects whose distance is less than k, we have $\delta_{\mathbf{S}}(b, c) - \delta_{\sigma(\mathbf{S})}(b, c) \le mk$.

Let s be a reachable state in $\mathcal{P}$, $\mathbf{S} = \langle \mathcal{B}, s \rangle$ its structure, and $\mathbf{G} = \langle \mathcal{B}, s \cup \psi_G \rangle$. By Corollary 11, the length of any $\sigma(s)$-plan is greater than or equal to $(\delta_{\sigma(\mathbf{S})}(b, c) - \delta_{\sigma(\mathbf{G})}(b, c))/(D - 1)$ for any $b, c \in \sigma(\mathcal{B})$ such that $\delta_{\sigma(\mathbf{G})}(b, c) < \infty$. The action diameters depend only on the action schemata $\mathcal{A}$. Thus they are not changed by σ. So the maximum diameter D is the same for $\mathcal{P}^+$ and $\sigma(\mathcal{P}^+)$. We have $\delta_{\sigma(\mathbf{S})}(b, c) \ge \delta_{\mathbf{S}}(b, c) - mk$ by Lemma 17 and $\delta_{\sigma(\mathbf{G})}(b, c) \le \delta_{\mathbf{G}}(b, c)$ by Lemma 5(2). Thus

$$\delta_{\sigma(\mathbf{S})}(b, c) - \delta_{\sigma(\mathbf{G})}(b, c) \ge \delta_{\mathbf{S}}(b, c) - \delta_{\mathbf{G}}(b, c) - mk$$

Hence the lower bound decreases at most by $mk/(D - 1)$ w.r.t. the lower bound for an s-plan in $\mathcal{P}^+$. To make k small, we must identify objects close to each other in any state reachable in the original PDDL task $\mathcal{P}$.

However, when creating σ, we only know the initial state $\mathbf{I} = \langle \mathcal{B}, \psi_I \rangle$. Consequently, the objects close in $\mathbf{I}$ need not be close in a reachable state in $\mathcal{P}$. To overcome this issue, we focus on static predicates. A predicate is said to be **static** if it appears only in action preconditions. The remaining predicates are called **dynamic** (they occur in an action effect). Thus we can split any state $s = s^S \cup s^D$

Table 1: Number of solved tasks by A* with the blind heuristic h^0, lifted $h^{\max}$, and h^{lmc} on tasks reduced by 25%, 50%, 75% and 95% of objects. Only domains where `rnd` and `gf` differ for at least one level of reduction are shown. Numbers in bold indicate that which of `rnd` or `gf` is better for the fixed reduction level.

domain			25%		50%		75%		95%	
	h^0	$h^{\max}$	rnd	gf	rnd	gf	rnd	gf	rnd	gf
blocks (65)	0	2	4	4	6	**8**	9	**14**	11	**34**
logistics (40)	4	10	**14**	13	16	**19**	20	**28**	27	**30**
organic-synthesis (56)	44	35	26	**31**	26	**32**	27	**32**	27	**32**
pipesworld-tankage (50)	11	12	7	**9**	12	**13**	15	15	15	**16**
rovers (40)	0	3	**11**	6	**10**	5	**7**	4	**5**	4
visitall (180)	32	86	63	**65**	60	**63**	59	**62**	59	**62**
others (556)	36	37	41	41	41	41	41	41	41	41
Σ (987)	127	185	166	**169**	171	**181**	178	**196**	185	**219**

where s^S represents static atoms and s^D dynamic atoms. In particular, for the initial state, we have $\psi_I = \psi_I^S \cup \psi_I^D$. Observe that for any reachable state s, it holds $s^S = \psi_I^S$ because the same relations interpret the static predicates in the whole reachable space. In other words, the structure $\mathbf{I}^S = \langle \mathcal{B}, \psi_I^S \rangle$ is a substructure of any reachable state. Hence it can be seen as a sort of fixed map, whereas the dynamic atoms connect objects to particular locations within this map. Using this optic, identifying close objects in $\mathbf{I}^S$ shortens the paths through the static map. This choice seems to be reasonable. If $\delta_{\mathbf{I}^S}(b,c)$ is small, then $\delta_{\mathbf{S}}(b,c)$ is small for all reachable states $\mathbf{S}$ because $\delta_{\mathbf{S}}(b,c) \leq \delta_{\mathbf{I}^S}(b,c)$ by Lemma 5(3).

To summarize the construction of σ, we iteratively identify a pair of objects until we reach the desired object reduction. To select a pair, we first try identifying pairs of objects that are close in $\mathbf{I}^S$. If there are no candidates (for instance, in the domains without static predicates like blocks), we identify objects close in the initial state $\mathbf{I} = \langle \mathcal{B}, \psi_I \rangle$.

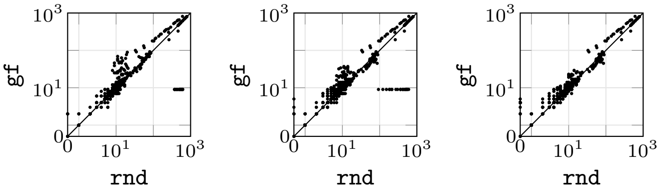
6 Experimental Evaluation

The proposed methods were implemented[4] in C and evaluated on a cluster with AMD EPYC 7543 processors and 30 minutes time and 4 GB memory limit for each process. We use the so-called hard-to-ground (HTG) domains [18, 8, 2, 16]. We merged together different variants of the same domain, and we removed domains with conditional effects, fully grounded domains, and duplicates, leaving 987 tasks in 12 domains.

The input PDDL tasks are pruned with PDDL endomorphisms [15] and with lifted fact-alternating mutex groups [5, 6]. The proposed method inferring homomorphisms based on Gaifman graphs is denoted by `gf`, and the method from [14] selecting homomorphisms randomly is denoted by `rnd`. Both `gf` and `rnd` fix goals and allow mapping between different objects only if they are of the same type. `rnd` generates up to 50 random homomorphisms within one minute time limit out of which the one resulting in the highest estimate for the initial state is selected. `gf` may need to break ties when deciding between multiple homomorphism. In such case, we select randomly among pairs of objects whose distance in the Gaifman graph is the same, and in this way we also generate up to 50 random homomorphisms within one minute time limit and select the one resulting in the highest heuristic estimate for the initial state.

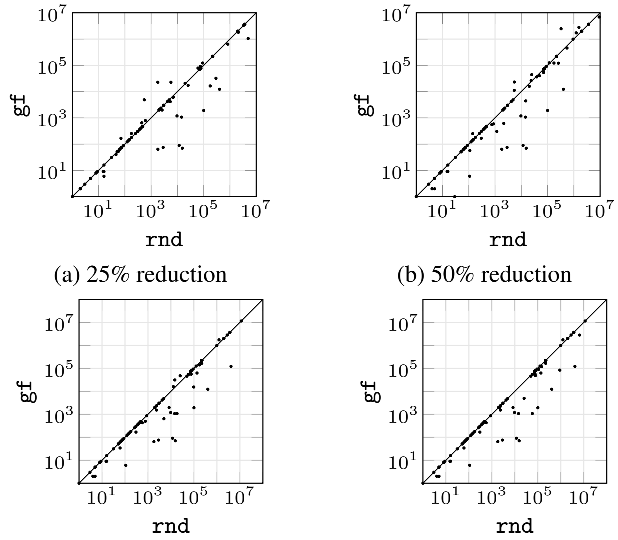
We evaluate `gf` and `rnd` in four variants—we reduce the number of objects by up to 25%, 50%, 75%, or 95%. When grounding the reduced tasks, we apply pruning using the h^2 heuristic [10] to make the ground tasks even smaller.

[4] https://gitlab.com/danfis/cpddl, branch ecai23-lifted-hmorph-gaifman



(a) 50% reduction (b) 75% reduction (c) 95% reduction

Figure 4: Per-task comparison of the h^{lmc} heuristic values for initial states for three best-performing variants of `gf` and `rnd`. Scatter plots use logarithmic scale with artificially added zero. Only tasks where both variants were able to compute the heuristic value are included.



(a) 25% reduction (b) 50% reduction (c) 75% reduction (d) 95% reduction

Figure 5: Per-task comparison of the number of expanded states before the last f-layer for commonly solved tasks by A* with h^{lmc}.

We apply these methods in both optimal and satisficing setting. In the optimal setting, we use A⋆ [9] as the search algorithm. For `gf` and `rnd`, we use the LM-Cut heuristic (h^{lmc}) [11] computed on the ground reduced tasks, and we compare them to the blind heuristic assigning zero to all states (h^0), and the lifted $h^{\max}$ heuristic computed using Datalog in each state ($h^{\max}$) [1]. In the satisficing setting, we use greedy best-first search instead of A⋆. For `gf` and `rnd`, we use the FF heuristic (h^{ff}) [13], and we compare them to the lifted Datalog-based h^{add} heuristic [1].

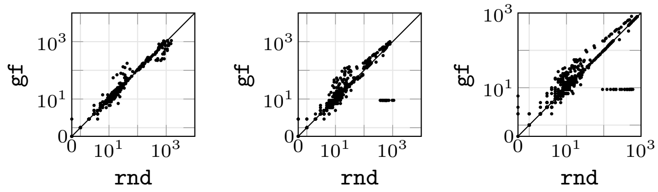
Table 1 summarizes the number of solved tasks in the optimal setting. The results suggest using Gaifman graphs, indeed, tends to be more informed than the random selection more often than not. The two best-performing variants of `gf` (75% and 95%) outperform $h^{\max}$ in the overall number of solved tasks, but the results vary across the tested domains.

Figure 4 compares `gf` and `rnd` in terms of the h^{lmc} heuristic values in initial states. One can observe that `gf` often results in a more informed heuristic than `rnd` under the same reduction level. However, we should note that these results may be misleading as the selection of the homomorphisms is steered towards selecting the ones maximizing heuristic values in initial states. That is, there is no guarantee that this behaviour is actually replicated in state other than the initial one.

To get a better sense how the methods compare in the A⋆ search,

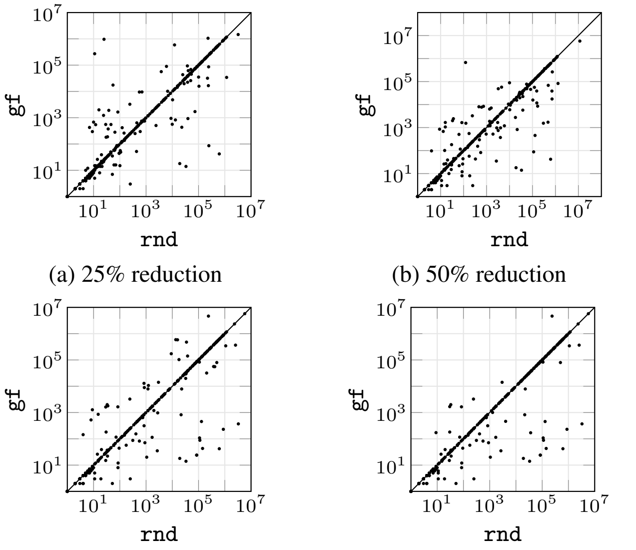
Table 2: Number of solved tasks as in Table 1, but for greedy best-first search with the lifted h^{add}, and h^{ff} for `gf` and `rnd`.

domain	h^{add}	25%		50%		75%		95%	
		rnd	gf	rnd	gf	rnd	gf	rnd	gf
agricola (25)	0	**3**	0	0	0	0	0	0	0
childsnack (144)	61	58	**60**	49	**56**	**47**	45	37	37
ged (312)	112	**125**	124	125	125	125	125	125	125
logistics (40)	31	18	**23**	11	**20**	10	**20**	5	**20**
organic-synthesis (56)	34	26	**32**	26	**32**	28	**32**	28	**32**
pipesworld-tankage (50)	22	**14**	13	12	**14**	14	14	13	**15**
rovers (40)	12	12	**13**	6	**10**	7	**10**	4	**7**
tpp (25)	0	2	2	0	**1**	0	**1**	0	**1**
visitall (180)	140	98	**115**	87	**105**	85	**94**	84	**94**
others (115)	1	1	1	1	1	3	3	9	9
Σ (987)	413	357	**383**	317	**364**	319	**344**	305	**340**



(a) 25% reduction (b) 50% reduction (c) 75% reduction

Figure 6: Per-task comparison of the h^{ff} heuristic values for initial states similar to Figure 4.



(a) 25% reduction (b) 50% reduction

(c) 75% reduction (d) 95% reduction

Figure 7: Per-task comparison of the overall number of expanded states for commonly solved tasks by greedy best-first search with h^{ff}.

we compare the number of expanded states before the last f-layer for tasks solved by both `gf` and `rnd` (Figure 5). The overall picture, again, seems to be favourable to `gf` as the lower number of expanded states indicates better informativeness of `gf`. So, Gaifman graphs seem to carry a useful information that can be utilized in the optimal planning. Next, we look at the satisficing setting.

Table 2 compares the number of solved tasks for the greedy best-first search. As in the previous case, using `gf` tends to result in a higher number of solved tasks than `rnd`. However, h^{add} is superior, in particular (but not exclusively) due to the visitall domain.

We also compare the h^{ff} heuristic values for initial states (Figure 6) which tend to be higher for `gf`, suggesting reductions with `gf` lose less information than with `rnd`. However, in this case, one has to keep in mind that h^{ff} is not an admissible heuristic, so a higher value does not necessarily imply it is closer to the perfect heuristic.

Figure 7 compares the overall number of expanded states for the tasks solved by both `gf` and `rnd`. Here, the overall picture is not as convincing as in the optimal setting. Nevertheless, one can still observe some improvement.

Overall, we think the results are encouraging as they show Gaifman graphs can provide useful information about planning tasks.

7 Conclusion

We apply the metric induced by Gaifman graphs to lifted planning. For PDDL tasks with finite action schema diameter, we derive a lower bound on the length of the shortest plan. The obtained results help us to improve moderately the delete-free relaxation heuristics based on PDDL homomorphisms, as shown by our experimental results. In our experiments, we also tried to use the lower bound from Corollary 11 directly as a heuristic value for PDDL tasks with unit cost actions. However, the obtained results suggest that it is not sufficiently informative and the results were essentially the same as for the blind search.

Our theoretical results have several limitations. First, unary predicates are irrelevant to Gaifman graphs. Consequently, if the goal ψ_G consists only of unary atoms, the lower bound on the plan length from Corollary 11 is trivially zero as adding ψ_G to a state does not change the metric. This situation happens, for example, in the visitall domain where the goal consists of atoms of the form $\texttt{visited}(l)$ for a location l. Nevertheless, it is possible to reformulate the domain so that the lower bound would be more informative by making `visited` a binary predicate connecting the initial location and the visited one.

A further limitation is the finite action diameter. Most of the action schemata in the IPC domains have finite diameters, but a few action schemata have infinite ones. This is, again, usually caused by the unary predicates. One such example is from the floortile domain. Its action schema `change-color` has preconditions $\texttt{robot-has}(r, c)$, $\texttt{available-color}(c')$. Thus there is no edge connecting c' with other objects. Even though our theoretical results do not apply to `change-color`, it is possible to reformulate the task equivalently by introducing a new static binary predicate `p` whose interpretation connects all the colors, and we expand the preconditions by $\texttt{p}(c, c')$.

Another example of an action schema with an infinite diameter is `grasp` from the barman domain. Its preconditions contain only unary predicates, namely $\texttt{on-table}(c)$, $\texttt{handempty}(h)$. This can be equivalently reformulated as well. We can introduce a new object t representing a table, and we replace the preconditions with $\texttt{on-table}(c, t)$, $\texttt{handempty}(h, t)$. The first atom expresses that c is on the table t, whereas the second one asserts that the hand h is operating over the table t.

We think that enriching Gaifman graphs in a domain-independent manner with these kinds of information not explicitly specified in the input PDDL formulation might strengthen the proposed analysis. Moreover, we could also enrich Gaifman graphs by labeling edges with predicate symbols inducing the connections between objects which would provide more fine-grained information about how or why objects relate to each other. Another interesting question is whether we can incorporate action costs into our analysis so that we would get lower bounds on costs of optimal plans rather than on their lengths.

Acknowledgements

Rostislav Horčík was supported by the AFOSR award FA9550-18-1-0097 and the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics". Daniel Fišer was supported by DFG Grant 389792660 as part of TRR 248 (CPEC, https://perspicuous-computing.science).

References

[1] Augusto B. Corrêa, Guillem Francès, Florian Pommerening, and Malte Helmert, 'Delete-relaxation heuristics for lifted classical planning', in *Proceedings of the 31st International Conference on Automated Planning and Scheduling (ICAPS'21)*, pp. 94–102, (2021).

[2] Augusto B. Corrêa, Florian Pommerening, Malte Helmert, and Guillem Francès, 'Lifted successor generation using query optimization techniques', in *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling (ICAPS'20)*, pp. 80–89, (2020).

[3] Augusto B. Corrêa, Florian Pommerening, Malte Helmert, and Guillem Francès, 'The FF heuristic for lifted classical planning', in *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI'22)*, pp. 9716–9723, (2022).

[4] Heinz-Dieter Ebbinghaus and Jörg Flum, *Finite model theory*, Perspectives in Mathematical Logic, Springer, 1995.

[5] Daniel Fišer, 'Lifted fact-alternating mutex groups and pruned grounding of classical planning problems', in *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI'20)*, pp. 9835–9842, (2020).

[6] Daniel Fišer, 'Operator pruning using lifted mutex groups via compilation on lifted level', in *Proceedings of the 33rd International Conference on Automated Planning and Scheduling (ICAPS'23)*, pp. 118–127, (2023).

[7] Haim Gaifman, 'On Local and Non-Local Properties', in *Proceedings of the Herbrand Symposium*, ed., J. Stern, volume 107 of *Studies in Logic and the Foundations of Mathematics*, 105–135, Elsevier, (1982).

[8] Daniel Gnad, Álvaro Torralba, Martín Ariel Domínguez, Carlos Areces, and Facundo Bustos, 'Learning how to ground a plan - partial grounding in classical planning', in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI'19)*, pp. 7602–7609, (2019).

[9] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107, (1968).

[10] Patrik Haslum and Hector Geffner, 'Admissible heuristics for optimal planning', in *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems (AIPS'00)*, pp. 140–149, (2000).

[11] Malte Helmert and Carmel Domshlak, 'Landmarks, critical paths and abstractions: What's the difference anyway?', in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09)*, pp. 162–169, (2009).

[12] Wilfrid Hodges, *A Shorter Model Theory*, Cambridge University Press, 1997.

[13] Jörg Hoffmann and Bernhard Nebel, 'The FF planning system: Fast plan generation through heuristic search', *Journal of Artificial Intelligence Research*, **14**, 253–302, (2001).

[14] Rostislav Horčík, Daniel Fišer, and Álvaro Torralba, 'Homomorphisms of lifted planning tasks: The case for delete-free relaxation heuristics', in *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI'22)*, pp. 9767–9775, (2022).

[15] Rostislav Horčík and Daniel Fišer, 'Endomorphisms of classical planning tasks', in *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*, pp. 11835–11843, (2021).

[16] Pascal Lauer, Alvaro Torralba, Daniel Fišer, Daniel Höller, Julia Wichlacz, and Jörg Hoffmann, 'Polynomial-time in PDDL input size: Making the delete relaxation feasible for lifted planning', in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI'21)*, pp. 4119–4126, (2021).

[17] Leonid Libkin, *Elements of Finite Model Theory*, Texts in Theoretical Computer Science. An EATCS Series, Springer, 2004.

[18] Arman Masoumi, Megan Antoniazzi, and Mikhail Soutchanski, 'Modeling organic chemistry and planning organic synthesis', volume 36 of *EPiC Series in Computing*, pp. 176–195, (2015).

[19] Drew McDermott, 'The 1998 AI planning systems competition', *The AI Magazine*, **21**(2), 35–55, (2000).