Compact Representatives of Potential Heuristics

Simon Dold, Malte Helmert
University of Basel, Basel, Switzerland
{simon.dold,malte.helmert}@unibas.ch

Abstract

There are uncountably many potential heuristics for a given classical planning task and a given set of features. However, if two potential heuristics are sufficiently similar they do not change the behavior of the search algorithm that uses them. For many satisficing planning approaches two heuristics are sufficiently similar if they provide the same weak ordering of the states.

In this work, we categorize all potential heuristics of a given task and set of features into equivalence classes. We show that each equivalence class contains one potential heuristic with a representation size polynomial in the number of features.

This shows that a non-deterministic Turing Machine is able to guess a potential heuristic from such an equivalence class in polynomial time and closes a small theoretical gap for the complexity of heuristic synthesis for satisficing classical planning Helmert et al. (2022) left.

Introduction

Satisficing classical planning is interested in reachability in a factored transition system induced by a compactly encoded problem description.It ignores the cost and length of the plan but optimizes only for search time. Most solvers participating in the satisficing track of the International Planning Competition use heuristic search.

Potential heuristics (Pommerening et al. 2015) are a flexible way to represent heuristics for classical planning. Many approaches for the synthesis of potential heuristics have been investigated (Pommerening et al. 2015; Seipp, Pommerening, and Helmert 2015; Seipp et al. 2016; Štolba, Fišer, and Komenda 2016; Francès et al. 2019). Recently, Helmert et al. (2022) looked at the complexity of synthesis of potential heuristics with special properties desirable for satisficing planning. These properties are implied by simple constraints. Helmert et al. (2022) restricted their results to the synthesis of *compact* heuristics. They used this restriction to allow polynomial guessing. We show that this induces no real restriction, because for each potential heuristic there is a compact one that is equivalent for their purposes.

Overview

First, we provide the necessary background. Afterwards, we define the equivalence classes of potential heuristics (Definition 4). Then, we show that each equivalence class has a compact representative (Theorem 10). We discuss the complexity results of Helmert et al. (2022) that are restricted to *compact* classes and show that this restriction can be relaxed to *succinct* classes (Theorem 15–17). Additionally, we show that many relevant classes are succinct (Theorem 19).With that we strengthen the results of Helmert et al. (2022) for the complexity of heuristic synthesis for satisficing classical planning (Corollary 23–25).

Background

For a compactly encoded planning task each state in the transition system is a full assignment of the **state variables** to their respective **domain values**. The set of state variables $V = \{v_1, \dots, v_n\}$ and their finite domains $dom(v_i)$ induce a *variable space* $\mathcal{V}$, with a representation size $\llcorner \mathcal{V} \lrcorner = \sum_{i=1}^{n} |dom(v_i)|$. We require a planning task to be encoded in a way that allows a quick successor generation i.e., an algorithm exists that generates all successors in time polynomially in the representation size of the task.

Heuristics

Detecting reachability in this setup is hard, especially for planning tasks with many state variables or many successors in individual states. To reduce search time, we use guidance that estimates how promising a given state is. We represent this guidance as a **heuristic** function h for the planning task Π that maps each state of Π to a value in $\mathbb{R} \cup \{\infty\}$, where smaller values represent more promising states. Note, that we allow the heuristic to map onto negative values in this work which is often not the case.

Potential heuristics are a class of heuristics introduced by Pommerening et al. (2015). A **potential heuristic** is a heuristic that is computed with a weighted count of the partial assignments (called **features**) that agree with the given state.

$$h^{pot}(s) = \sum_{f \in \mathcal{P}} w(f) \cdot [f \subseteq s]$$

where $\mathcal{P}$ is the set of all possible partial assignments for the task, $[f \subseteq s]$ is in the Iverson bracket notation, and $w(f) \in \mathbb{R} \cup \{\infty\}$ is the **weight** for the partial assignment f. In practice most of the weights are 0. The **dimension** of a potential heuristic is $max_{f \in \mathcal{P}, w(f) \neq 0} |f|$. Potential heuristics of small dimension with small weights are quick to evaluate but finding good weights that make the potential heuristic informed is complex.

Equivalence Classes

In satisficing planning algorithms like simple hill-climbing, the exact heuristic value of a state is not of importance, but only the comparison to other states. A heuristic h' that is a positive scaling of h, meaning $h'(s) := c \cdot h(s)$ with $c \in \mathbb{R}_{>0}$ causes the same behavior in the algorithm. There are other changes possible that guarantee to not affect the behavior as long as the heuristics h' and h are *qualitative equivalent*.

Definition 1 (Weak-Order Equivalent Heuristics)**.** We call heuristics h and h' on planning task Π **weak-order equivalent** if for any two states $s, s' \in \Pi$ it holds $h(s) > h(s')$ iff $h'(s) > h'(s')$.

Definition 2 (Sign Equivalent Heuristics)**.** We call heuristics h and h' on planning task Π **sign equivalent** if for any state $s \in \Pi$ it holds $h(s) > 0$ iff $h'(s) > 0$.

Definition 3 (Infinity Equivalent Heuristics)**.** We call heuristics h and h' on planning task Π **infinity equivalent** if for any state $s \in \Pi$ it holds $h(s) = \infty$ iff $h'(s) = \infty$.

Definition 4 (Qualitative Equivalent Heuristics)**.** We call heuristics h and h' on planning task Π **qualitative equivalent** if they are weak-order equivalent, sign equivalent and infinity equivalent.

The qualitative equivalence is clearly an equivalence relation (it is reflexive, symmetric and transitive). It induces the equivalence classes we consider.

For potential heuristics we are interested in changes to the non-zero weights that preserve the equivalence class. In the following we often indicate the **o**riginal heuristic with h^o and its equipotent **r**eduction with h^r. For a potential heuristic h, we define the set of features with non-zero weights $F(h) := \{f \in \mathcal{P} \mid w(f) \neq 0\}$.

Definition 5 (Equipotent Reduction)**.** Let h^o and h^r be potential heuristics on planning task Π with weight function w^o and w^r respectively. We call h^r an **equipotent reduction** of h^o if they are qualitative equivalent and $F(h^r) \subseteq F(h^o)$.

The equipotent reduction is clearly a transitive relationship.

Encode Equipotent Reductions as Linear Program

For a potential heuristic h^o, we define a linear program $L(h^o)$ with a set of constraints where each element $\overrightarrow{w}$ of the solution space represents a weight function w such that the corresponding potential heuristic is an equipotent reduction of h^o.

Similarly to $F(h)$, we define $F_{\mathit{fin}}(h) := \{f \in \mathcal{P} \mid w(f) \notin \{0, \infty\}\}$ and $F_{\mathit{inf}}(h) := \{f \in \mathcal{P} \mid w(f) = \infty\}$. In order to represent a potential heuristic as a single vector $\overrightarrow{w}$, we choose an arbitrary order $\mathit{idx} : F_{\mathit{fin}}(h) \to \{1, \dots, |F_{\mathit{fin}}(h)|\}$ for the partial states in $F_{\mathit{fin}}(h)$ and define the vector $\overrightarrow{w} \in \mathbb{R}^{|F_{\mathit{fin}}(h)|}$ with $\overrightarrow{w}[\mathit{idx}(f)] = w(f)$ for each $f \in F_{\mathit{fin}}(h)$. For simplicity, we sometimes write f instead of $\mathit{idx}(f)$ when clear from the context.

To extract the weight function w from such a vector $\overrightarrow{w}$ we set

$$w(f) := \begin{cases} \infty & \text{if } f \in F_{\mathit{inf}}(h^o) \\ \overrightarrow{w}[f] & \text{if } f \in F_{\mathit{fin}}(h^o) \\ 0 & \text{otherwise.} \end{cases}$$

By definition h^o is a potential heuristic, thus there exists a weight function w^o such that $h^o(s) = \sum_{f \in \mathcal{P}} w^o(f) \cdot [f \subseteq s]$ for each state s of task Π. We know that for each equipotent reduction h of h^o all weights $w(f)$ with $f \in \mathcal{P} \setminus F(h^o)$ are zero. Therefore, the sum can be reduced to $h(s) = \sum_{f \in F(h^o)} w(f) \cdot [f \subseteq s]$. For states with $h^o(s) = \infty$ we can reduce the sum to $h^o(s) = \sum_{f \in F_{\mathit{inf}}(h^o)} w(f) \cdot [f \subseteq s]$ and for states with $h^o(s) \neq \infty$ we can reduce the sum to $h^o(s) = \sum_{f \in F_{\mathit{fin}}(h^o)} w(f) \cdot [f \subseteq s]$.

Encode Infinity Equivalence The way we extract the weight function w from the vector $\overrightarrow{w}$ enforces the infinity equivalence. This also partially captures the sign equivalence and the weak-order equivalence, namely for those states where $h^o(s) = \infty$. Thus, we do not have to add constraints about such states.

Encode Sign Equivalence To enforce the sign equivalence we add the linear constraint

$$\sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s] \geq 1 \tag{1}$$

for each state s with $\infty \neq h^o(s) > 0$.

To prevent inequalities that would violate the sign equivalence we add the linear constraint

$$\sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s] \leq 0 \tag{2}$$

for each state s with $h^o(s) \leq 0$.

Encode Weak-Order Equivalence To enforce the inequalities necessary for the weak-order equivalence we add the linear constraint

$$\sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s] \geq 1 + \sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s'] \tag{3}$$

for each pair s, s' with $\infty \neq h^o(s) > h^o(s')$.

To prevent inequalities that would violate the weak-order equivalence we add the linear constraint

$$\sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s] \leq \sum_{f \in F_{\mathit{fin}}(h^o)} \overrightarrow{w}[f] \cdot [f \subseteq s'] \tag{4}$$

for each pair s, s' with $h^o(s) \not> h^o(s') \neq \infty$.

Encode Bounds The solution space is unbounded. To cover that we add the linear constraints

$$\overrightarrow{w}[f] \cdot \mathit{sign}(\overrightarrow{w^o}[f]) \geq 0 \tag{5}$$

for each $f \in F_{\mathit{fin}}(h^o)$, and optimize in the direction of $-\overrightarrow{w^o}$ to force the sign of $\overrightarrow{w}[f]$ to be the same as the sign of $\overrightarrow{w^o}[f]$ or 0. This implies that $\overrightarrow{0}$ is a basic solution and optimal if it is feasible. With that the construction of $L(h^o)$ is complete.

Clearly, each solution $\overrightarrow{w}$ encodes an equipotent reduction h^r of h^o. However, we have to reason that the solution space is not empty.

Lemma 6. For each potential heuristic h^o the solution space of $L(h^o)$ is not empty.

Proof. In the special case $\overrightarrow{w} \in \mathbb{R}^0 = \{\epsilon\}$, i.e. all weights of h^o are in $\{0, \infty\}$, we have the empty vector ϵ as a solution.

Otherwise, we prove this by constructing a solution. If $\overrightarrow{w^b}$ is not in the solution space, there has to be at least one constraint C that is violated. By construction, C is not one of the constraints from (2) nor (4) nor (5). If C is a constraint of the form (1) that encodes the sign equivalence, then there must be a state s where $1 > h^o(s) > 0$. In that case we define $\delta := h^o(s)$. Otherwise, C is a constraint of the form (3) that encodes the necessary inequalities for the weak-order equivalence and there must be a pair of states s, s' with $1 > h^o(s) - h^o(s') > 0$. In that case we define $\delta := h^o(s) - h^o(s')$. In both cases we consider the scalar multiplication $\frac{1}{\delta} \cdot \overrightarrow{w^b}$. This result satisfies all constraints that $\overrightarrow{w^b}$ satisfies and additionally C. As there is a finite number of constraints we can repeat this process finitely often until all constraints are satisfied. □

From the fundamental theorem of linear programming we know that if a solution exists, then there exists an optimal solution on a corner of the solution space (Thie and Keough 2008).

A corner of our solution space is the intersection of (at least) $|F_{\mathit{fin}}(h^o)|$-many constraints. We chose an arbitrary optimal solution on a corner and call the set of (tight) constraints that induce this corner $\mathcal{C}(h^o)$ (if there are more than $|F_{\mathit{fin}}(h^o)|$ constraints, we choose arbitrary $|F_{\mathit{fin}}(h^o)|$-many linearly independent ones). Changing the constraints of $\mathcal{C}(h^o)$ from non-strict inequalities to equalities produces a system of linear equations of the form $A\overrightarrow{x} = \overrightarrow{b}$ that we call $E(h^o)$. Note that $A \in \{1, 0, -1\}^{|F_{\mathit{fin}}(h^o)| \times |F_{\mathit{fin}}(h^o)|}$ and $\overrightarrow{b} \in \{0, 1\}^{|F_{\mathit{fin}}(h^o)|}$.

Compact Representative

A potential heuristic has a weight from $\mathbb{R}$ for each partial state. For some values of $\mathbb{R}$ it is impossible to represent them in finite memory. This raises the question whether there is an equipotent reduction h^r of an arbitrary potential heuristic h^o with weights in $\mathbb{R}$ such that all weights from h^r are *compact*. In this case compact means polynomial in the number of weights. The answer is yes.

Hadamard's inequality (Hadamard 1893), the Laplace expansion (Laplace 1772), and Cramer's rule (Cramer 1750) collectively imply that the system of linear equations $E(h^o)$ has a compact solution. For a modern presentation of these classical concepts, we refer to Horn and Johnson (2013).

With Hadamard's inequality we bound the determinant.

Lemma 7. If $A \in \mathbb{R}^{n \times n}$, each entry of A is of absolute value at most m and $n > 0$, then $|\det(A)| \leq m^n \cdot n^{n/2}$

Proof. Consider Hadamard's inequality

$$|\det(A)| \leq \textstyle\prod_{i=1}^{n} ||a_i||_2$$

where a_i is the i-th row of A. We know $||a_i||_2 \leq \sqrt{n \cdot (m^2)}$, thus $|\det(A)| \leq ((n \cdot m^2)^{1/2})^n = m^n \cdot n^{n/2}$. □

This is a tight bound, i.e. for some matrices this is an equality. One special case of it is particularly useful for us.

Corollary 8. If $A \in \{1, 0, -1\}^{n \times n}$ and $n > 0$, then $|\det(A)| \leq n^{n/2}$.

We consider the Laplace expansion (or cofactor expansion),

$$\det(A) = \textstyle\sum_{j=1}^{n} (-1)^{i+j} \cdot A[i, j] \cdot \det(A_{i,j}),$$

where $A[i, j]$ is the entry at row i and column j in A, and $A_{i,j}$ is the submatrix of A created by removing row i and column j. It follows that the determinant of an integer matrix is an integer.

Corollary 9. If $A \in \mathbb{Z}^{n \times n}$, then $\det(A) \in \mathbb{Z}$.

With that we can use Cramer's rule to show that each equivalence class has a compact representative.

Theorem 10. Each potential heuristic $h^{\mathbb{R}}$ with n non-zero weights on a planning task Π has an equipotent reduction $h^{\mathbb{Z}}$ where each finite weight from $h^{\mathbb{Z}}$ is in $\mathbb{Z}$ and of absolute value at most $n^{n/2}$.

Proof. We know that a solution to $A\overrightarrow{x} = \overrightarrow{b}$ from $E(h^{\mathbb{R}})$ induces an equipotent reduction of $h^{\mathbb{R}}$.

We consider Cramer's rule:

$$\overrightarrow{x}[i] = \frac{\det(A_i)}{\det(A)}$$

for $i \in [1, n]$ where A_i is the matrix A but with the i-th column replaced by $\overrightarrow{b}$. From Corollary 9 we know $\det(A_i)$ is an integer. The weight vector $\overrightarrow{w^{\mathbb{Q}}}[f] := \frac{\det(A_{\mathit{idx}(f)})}{\det(A)}$ for all features in $F_{\mathit{fin}}(h^{\mathbb{R}})$ induces an equipotent reduction $h^{\mathbb{Q}}$ of $h^{\mathbb{R}}$. Next, we consider the alternative weight vector $\overrightarrow{w^{\mathbb{Z}}}[f] = \overrightarrow{w^{\mathbb{Q}}}[f] \cdot \det(A)$. Since this corresponds to a positive scaling of the equipotent reduction of $h^{\mathbb{R}}$ induced by $\overrightarrow{w^{\mathbb{Q}}}$, the weight vector $\overrightarrow{w^{\mathbb{Z}}}$ induces an equipotent reduction of $h^{\mathbb{R}}$, too. From Corollary 8, we know that $\det(A_i) \leq n^{n/2}$. By definition, $w^{\mathbb{Z}}(f) = \det(A_{\mathit{idx}(f)})$ for all features in $F_{\mathit{fin}}(h^{\mathbb{R}})$. Thus, each finite weight of $h^{\mathbb{Z}}$ induced by $\overrightarrow{w^{\mathbb{Z}}}$ has absolute value at most $n^{n/2}$. □

The question of existence of a compact solution for $E(h^{\mathbb{R}})$ can be answered positively by referring to Bottesch et al. (2020). However, their bound is $n!$ instead of $n^{n/2}$ and therefore less tight.

Discussion

Helmert et al. (2022) investigated the complexity of the synthesis of potential heuristics with properties desirable for satisficing planning. In their work, they considered different *classes* of potential heuristics.

Similarly to Helmert et al. (2022), we define the representation size of a potential heuristic h with weights in $\mathbb{Z} \cup \{\infty\}$ as $\llcorner h \lrcorner = |F(h)| \cdot \max_{f \in F(h)}(|f| + \lceil \log_2(|w(f)|) \rceil + 2)$, with the special case $\log_2(\infty) = 1$ i.e. we can represent ∞ with a special symbol. We define the representation size of a planning task $\llcorner \Pi \lrcorner$ to be at least the total number of atoms. Different but polynomially equivalent encodings of potential heuristics and planning tasks would not change the complexity results.

Definition 11 (Potential Heuristic Class (Helmert et al. 2022))**.** A **potential heuristic (PH) class** $\mathcal{C}$ is a function that maps any planning task Π to a family $\mathcal{C}(\Pi)$ of potential heuristics on Π, where $h \in \mathcal{C}(\Pi)$ can be tested in polynomial time in $\llcorner h \lrcorner$ and $\llcorner \Pi \lrcorner$.

Definition 12 (Helmert et al., 2022, Def. 2)**.** $\textsc{Synthesis}(\mathcal{P}, \mathcal{C})$ is the following decision problem: given a planning task Π, does there exist a potential heuristic $h^{\text{pot}} \in \mathcal{C}(\Pi)$ that has property $\mathcal{P}$ for Π?

For many of their membership results, they restrict themselves to *compact* PH classes.

Definition 13 (Compact Class (Helmert et al. 2022))**.** A PH class $\mathcal{C}$ is **compact** if there exists a polynomial p with $\llcorner h \lrcorner \leq p(\llcorner \Pi \lrcorner)$ for each task Π and each heuristic $h \in \mathcal{C}(\Pi)$.

This requirement is quite strict. Motivation for their restriction to compact PH classes is the idea of quickly "guessing" a potential heuristic with the property if one exists. Such a quick guess is only possible if we can exclude the edge case that all correct guesses have a too large representation size. Here, "quick" means polynomially in the representation size of the task.

However, it is not necessary that all heuristics in one PH class have a polynomially bounded representation size. It is sufficient if it is guaranteed that one with the considered property exists. Then we can still quickly guess a correct one. For this, we define a new type of PH classes based on qualitative equivalence, since all properties considered by Helmert et al. (2022) for synthesis either hold for each or no heuristic of the same equivalence class.

Definition 14 (Succinct Class)**.** A class is **succinct** if there exists a polynomial p such that for all tasks Π and $h \in \mathcal{C}(\Pi)$ there is an $h' \in \mathcal{C}(\Pi)$ that is qualitative equivalent to h with $\llcorner h' \lrcorner \leq p(\llcorner \Pi \lrcorner)$.

Their results on compact PH classes extend to succinct PH classes, because in both we can guess a correct member in polynomial time. With that, we can strengthen some of their synthesis membership results. We list the theorems we extend from Helmert et al. (2022) here.

Theorem 15 (extends Helmert et al. (2022), Thm. 4)**.** $\textsc{Synthesis}(\text{DDA}, \mathcal{C}) \in$ **PSPACE** for all succinct PH classes $\mathcal{C}$.

Theorem 16 (extends Helmert et al. (2022), Thm. 7)**.** $\textsc{Synthesis}(\text{SDDA}, \mathcal{C}) \in$ **PSPACE** for all succinct PH classes $\mathcal{C}$.

Theorem 17 (extends Helmert et al. (2022), Thm. 9)**.** $\textsc{Synthesis}(\text{VDDA}, \mathcal{C}) \in \mathbf{\Sigma}_2^{\text{p}}$ for all succinct PH classes $\mathcal{C}$.

In their work they specifically considered PH classes with bounded feature size.

Definition 18 (Helmert et al., 2022, Def. 3)**.** For $k \in \mathbb{N}_0 \cup \{\infty\}$, $\mathcal{C}^k$ denotes the PH class that allows all potential heuristics of dimension at most k. (In particular, $\mathcal{C}^\infty$ imposes no restrictions.)

With the results we presented in the previous section, we see that for finite k these classes are succinct.

Theorem 19. For $k \in \mathbb{N}_0$, $\mathcal{C}^k$ is a succinct PH class.

Proof. For each heuristic $h \in \mathcal{C}^k(\Pi)$ of any task Π the amount of non-zero weight features is bounded by the number of k-sized features i.e. $\llcorner \mathcal{V} \lrcorner^k$. By Theorem 10, there is a heuristic h' that is qualitative equivalent to h where the largest absolute value for each finite feature weight of h' is bounded by $\llcorner \mathcal{V} \lrcorner^{k \cdot \llcorner \mathcal{V} \lrcorner^k / 2}$.

With that, we see $\llcorner h' \lrcorner \leq \llcorner \mathcal{V} \lrcorner^k \cdot (k + \lceil \log_2(\llcorner \mathcal{V} \lrcorner^{k \cdot \llcorner \mathcal{V} \lrcorner^k / 2}) \rceil + 2)$. We conclude $\llcorner h' \lrcorner \in \mathcal{O}((k+1) \cdot \llcorner \mathcal{V} \lrcorner^{2k+2})$. Thus, for each $k \in \mathbb{N}_0$, there is a degree $2k+2$ polynomial $p \in \mathcal{O}((k+1) \cdot \llcorner \mathcal{V} \lrcorner^{2k+2})$ with $\llcorner h' \lrcorner \leq p(\llcorner \mathcal{V} \lrcorner) \leq p(\llcorner \Pi \lrcorner)$ for each task Π. $\square$

Next, we list some hardness results from Helmert et al. (2022). With these and the ones we just established we can provide novel completeness results.

Theorem 20 (Helmert et al., 2022, Thm. 5)**.** $\textsc{Synthesis}(\text{DDA}, \mathcal{C}^k)$ is **PSPACE**-hard for all $k \in \mathbb{N}_1$.

This result is not explicitly stated in their work but implicitly covered by their discussion after Thm. 5. There they focused on $\mathcal{C}^1$ and $\mathcal{C}^2$ as it was an open problem at that time whether solvable planning tasks exists where no DDA heuristic in $\mathcal{C}^k$ exists for $k > 3$. That this is the case for each $k \in \mathbb{N}$ was later shown by Dold and Helmert (2024). However, their argument works for each $\mathcal{C}^k$ if a solvable task Π' with no DDA heuristic in $\mathcal{C}^{k-1}$ exists.

Theorem 21 (Helmert et al., 2022, Thm. 7)**.** $\textsc{Synthesis}(\text{SDDA}, \mathcal{C}^k)$ is **PSPACE**-hard for all $k \in \mathbb{N}_1$.

Theorem 22 (Helmert et al., 2022, Thm. 10)**.** $\textsc{Synthesis}(\text{VDDA}, \mathcal{C}^k)$ is $\mathbf{\Sigma}_2^{\text{p}}$-hard for all $k \in \mathbb{N}_1$.

Conclusion

With the concept of qualitative equivalence and succinct PH classes we were able to extend the membership results of Helmert et al. (2022) from compact classes to succinct classes. This is in itself a minor step. With the LP encoding for equipotent reduction we showed that the PH class $\mathcal{C}^k$ is succinct. The combination of these two steps provides completeness results that were previously unavailable. We conclude with the completeness results that emerge.

Corollary 23. $\textsc{Synthesis}(\text{DDA}, \mathcal{C}^k)$ is **PSPACE**-complete for all $k \in \mathbb{N}_1$.

Proof. Follows by combining Theorems 15, 19 and 20. $\square$

Corollary 24. $\textsc{Synthesis}(\text{SDDA}, \mathcal{C}^k)$ is **PSPACE**-complete for all $k \in \mathbb{N}_1$.

Proof. Follows by combining Theorems 16, 19 and 21. $\square$

Corollary 25. $\textsc{Synthesis}(\text{VDDA}, \mathcal{C}^k)$ is $\mathbf{\Sigma}_2^{\text{p}}$-complete for all $k \in \mathbb{N}_1$.

Proof. Follows by combining Theorems 17, 19 and 22. $\square$

Acknowledgments

We thank Gustavo Delazeri for proofreading this paper and providing helpful feedback.

This research was supported by the Swiss National Science Foundation (SNSF) as part of the project "Unifying the Theory and Algorithms of Factored State-Space Search" (UTA).

References

Bottesch, R.; Haslbeck, M. W.; Reynaud, A.; and Thiemann, R. 2020. Verifying a Solver for Linear Mixed Integer Arithmetic in Isabelle/HOL. In *Proceedings of the 12th NASA Formal Methods Symposium (NFM 2020)*, volume 12229 of *Lecture Notes in Computer Science*, 233–250. Cham: Springer.

Cramer, G. 1750. *Introduction à l'analyse des lignes courbes algébriques*. Geneva: Chez les Frères Cramer & Cl. Philibert. Cramer's Rule is presented in Appendix No. I, pp. 657–659.

Dold, S.; and Helmert, M. 2024. Higher-Dimensional Potential Heuristics: Lower Bound Criterion and Connection to Correlation Complexity. In *Proc. ICAPS 2024*, 151–161.

Francès, G.; Corrêa, A. B.; Geissmann, C.; and Pommerening, F. 2019. Generalized Potential Heuristics for Classical Planning. In *Proc. IJCAI 2019*, 5554–5561.

Hadamard, J. 1893. Résolution d'une question relative aux déterminants. *Bulletin des sciences mathématiques*, 17: 240–246.

Helmert, M.; Sievers, S.; Rovner, A.; and Corrêa, A. B. 2022. On the Complexity of Heuristic Synthesis for Satisficing Classical Planning: Potential Heuristics and Beyond. In *Proc. ICAPS 2022*, 124–133.

Horn, R. A.; and Johnson, C. R. 2013. *Matrix Analysis*. New York, NY: Cambridge University Press, second edition. First published 1985. Second edition published 2013.

Laplace, P.-S. 1772. *Recherches sur le calcul intégral et sur le système du monde*, volume 8, 369–477. Paris: Gauthier-Villars. Expansion of determinants discussed on pp. 395–404. Original work presented 1772, published 1776.

Pommerening, F.; Helmert, M.; Röger, G.; and Seipp, J. 2015. From Non-Negative to General Operator Cost Partitioning. In *Proc. AAAI 2015*, 3335–3341.

Seipp, J.; Pommerening, F.; and Helmert, M. 2015. New Optimization Functions for Potential Heuristics. In *Proc. ICAPS 2015*, 193–201.

Seipp, J.; Pommerening, F.; Sievers, S.; Wehrle, M.; Fawcett, C.; and Alkhazraji, Y. 2016. Fast Downward Aidos. In *Unsolvability IPC: Planner Abstracts*, 28–38.

Thie, P. R.; and Keough, G. E. 2008. *An introduction to linear programming and game theory (3rd edition)*. John Wiley & Sons.

Štolba, M.; Fišer, D.; and Komenda, A. 2016. Potential Heuristics for Multi-Agent Planning. In *Proc. ICAPS 2016*, 308–316.