Frameworks for logically classifying polynomial-time optimisation problems

James Gate and Iain A. Stewart
School of Engineering and Computing Sciences,
Durham University, Science Labs,
South Road, Durham DH1 3LE, U.K.
e-mail: j.s.gate/i.a.stewart@durham.ac.uk

November 19, 2009

Abstract

We show that a logical framework, based around a fragment of existential second-order logic formerly proposed by others so as to capture the class of polynomially-bounded P-optimisation problems, cannot hope to do so, under the assumption that $\mathsf{P} \neq \mathsf{NP}$. We do this by exhibiting polynomially-bounded maximisation and minimisation problems that can be expressed in the framework but whose decision versions are NP-complete. We propose an alternative logical framework, based around inflationary fixed-point logic, and show that we can capture the above classes of optimisation problems. We use the inductive depth of an inflationary fixed-point as a means to describe the objective functions of the instances of our optimisation problems.
keywords: finite model theory; descriptive complexity; optimisation problems.

1 Introduction

The theory of computational complexity is primarily concerned with the classification of decision problems, and although many (NP-complete) decision problems are actually decision versions of more natural optimisation problems, the classification of optimisation problems does not fit naturally into many of the available standard classification frameworks. Whilst there do exist criteria against which we can classify optimisation problems, such as according to their approximation properties [11], it was not until Papadimitriou and Yannakakis [18] proposed the use of existential second-order logic as a means for classification that a natural and robust framework became available. The classification of optimisation problems within this logical framework was subsequently significantly clarified by Panconesi and Ranjan [17] and Kolaitis and Thakur [13, 14] (we briefly explain Kolaitis and Thakur's work later).

The optimisation problems considered in the papers above are (polynomially-bounded) NP-optimisation problems. The class of (polynomially-bounded) P-optimisation problems is an important sub-class of optimisation problems. Typical of P-optimisation problems are the maximum unit flow problem, the maximum 2-satisfiability problem, the minimum shortest-path problem and the minimum cut problem. In [16], Manyem attempted to logically capture the class of polynomially-bounded P-optimisation problems by utilizing a fragment of existential second-order logic (where the first-order part of any formula is a universally-quantified Horn formula) which is known to capture the class of decision problems P (on ordered structures; this characterization was due to Grädel [6]). As we demonstrate here, proceeding as Manyem suggests results in failure (assuming that $\mathsf{P} \neq \mathsf{NP}$), for there are polynomially-bounded optimisation problems that can be expressed within his logical framework but whose associated decision problems are NP-complete (by definition, an NP-optimisation problem is a P-optimisation problem if its associated decision problem is in P). However, we present a new framework based around inflationary fixed-point logic and where we use the inductive depth of an inflationary fixed-point as a mechanism by which to compute the values of the objective functions of the instances of our optimisation problems. We show that the classes of polynomially-bounded P-maximisation problems and polynomially-bounded P-minimisation problems can be captured within our framework.

In the next section we give definitions as regards optimisation problems relevant to this paper, we explain Kolaitis and Thakur's logical frameworks and their main results, and we present Manyem's logical framework for the attempted characterization of classes of polynomially-bounded P-optimisation problems. In Section 3, we prove that Manyem's logical framework does not yield logical characterizations of the classes of polynomially-bounded P-maximisation and polynomially-bounded P-minimisation problems, before presenting a new logical framework in Section 4 which enables us to logically characterize these classes of optimisation problems. Our conclusions and directions for further research are presented in Section 5.

2 A framework for classification

In this section, we define classes of (non-deterministic polynomial-time) optimisation problems and provide logical frameworks for the classification of such problems. These classes and frameworks come from [13, 14, 17, 18]. Furthermore, we present some classification results from [13, 14]. In addition, we refine definitions and notions from [1, 16]. In particular, we define classes of (deterministic) polynomial-time optimisation problems and we explain how the logical framework presented in [1, 16], together with the subsequent analysis, was somewhat incongruous.

2.1 P-optimisation problems

We begin by defining what we mean by a polynomial-time optimisation problem, or P-optimisation problem for short.

Definition 1 A *maximisation problem* (resp. *minimisation problem*) $\mathcal{Q}$ is a 4-tuple $(\mathcal{I}, \mathcal{F}, f, opt)$ where:

1. $\mathcal{I}$ is the set of instances of $\mathcal{Q}$, with $\mathcal{I}$ recognisable in polynomial-time;

2. $\mathcal{F}$ is the set of feasible solutions to some instance of $\mathcal{I}$, where we denote by $\mathcal{F}(I)$ the set of feasible solutions to instance I;

3. $f : \mathcal{I} \times \mathcal{F} \to \mathbb{N} \cup \{\bot\}$ is the objective function, and is such that:

   - $f(I, J) = \bot$ if, and only if, $J \notin \mathcal{F}(I)$;
   - there is a polynomial p_f such that $f(I, J)$ is computable in time $p_f(|I|)$;

4. for any instance $I \in \mathcal{I}$, if $\mathcal{F}(I)$ is non-empty then $opt(I) = max\{f(I, J) : J \in \mathcal{F}(I)\}$ (resp. $opt(I) = min\{f(I, J) : J \in \mathcal{F}(I)\}$), and if $\mathcal{F}(I)$ is empty then $opt(I) = \bot$.

The class of *optimisation problems* consists of the class of maximisation problems in union with the class of minimisation problems. The maximisation (resp. minimisation) problem $\mathcal{Q}$ is a P-*maximisation problem* (resp. P-*minimisation problem*) if:

5. the problem of deciding whether a given instance $I \in \mathcal{I}$ and a given integer m are such that there exists a feasible solution $J \in \mathcal{F}(I)$ such that $f(I, J) \geq m$ (resp. $f(I, J) \leq m$) can be solved in polynomial-time.

The class of P-*optimisation problems* P_{opt} is the class of P-maximisation problems P_{max} in union with the class of P-minimisation problems P_{min}. The classes of NP-*optimisation problems* NP_{opt}, NP-*maximisation problems* NP_{max}, and NP-*minimisation problems* NP_{min} are defined analogously except that in condition 5 'non-deterministic polynomial-time' replaces 'polynomial-time'. We refer to the problem in condition 5 as the *decision version* of $\mathcal{Q}$. □

Importantly, for us the *solution* of an optimisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ is an algorithm that given any instance I of the problem, produces as output the value $opt(I)$ and *not* (necessarily) an optimal feasible solution from $\mathcal{F}(I)$ (if there is one). In fact, this algorithm need not even work with representations of feasible solutions; all it has to do is to come up with the optimal value. Note that all problems $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ in P_{opt} can be solved in polynomial-time, for: given any instance I of size n and any feasible solution $J \in \mathcal{F}(I)$, by definition $f(I, J)$ is $O(2^{p(n)})$, where p is some polynomial; and repeating the algorithm in condition 5 in tandem with a binary search yields a polynomial-time algorithm that computes $opt(I)$.

We have three remarks.

Remark 2 Note that Definition 1 implies that all feasible solutions to some instance can be taken to have size bounded by some polynomial in the size of the instance. Given that our notion of a solution of an optimisation problem is such that a numeric value should be found and not a witnessing feasible solution, there is no real need to discuss the computational nature of a set of feasible solutions corresponding to some instance. In particular, Definition 1 says nothing about the complexity of deciding whether some potential feasible solution is indeed an actual feasible solution to some instance. It turns out that most (instances of) natural optimisation problems have easily recognizable sets of feasible solutions.

Remark 3 The reader will have noted that according to Definition 1, *every* optimisation problem is in fact an NP-optimisation problem, and so condition 5 is redundant when defining an NP-optimisation problem. However, we have included it as it appears in analogous definitions in [13, 14]; for in these definitions the objective function is defined to be computable in time polynomial in the size of the input, i.e., the instance and a feasible solution, rather than in the size of the instance. Our notion of an optimisation problem is such that every feasible solution to some instance necessarily has size bounded by some polynomial in the size of the instance, whereas in [13, 14] there is scope for considering optimisation problems whose instances have feasible solutions of size exponential in the size of the instance. As far as we know, such optimisation problems have never been considered.

Remark 4 We should point out that Manyem's definition of a P-optimisation problem in [16], and subsequently in [1], is slightly different from that in Definition 1, for Manyem had an extra condition, namely that:

6′. the problem of computing an optimal solution for a given instance I of $\mathcal{Q}$ can be solved in time polynomial in $|I|$.

We have dropped this condition as we feel that the condition is not intrinsic to our notion of the solution of an optimisation problem (for us, and as is the case in [13, 14], we are only concerned with whether solutions of certain sizes exist and not with actually exhibiting such solutions). We wish our P-optimisation problems to be analogous to the NP-optimisation problems of [13, 17, 18] where no such condition exists. Indeed, imposing such a condition in the context of NP is somewhat problematic as not only would we be asking for a non-deterministic polynomial-time transducer but checking optimality would provide difficulties too. Dropping Manyem's additional condition provides for a more appropriate analysis. (Note that the conditions 1 – 5 and 6′, above, mirror Manyem's conditions (i) – (vi) in [16].)

2.2 Polynomially-bounded P-optimisation problems

The classes defined in the following definition play an important role (as we shall explain soon).

Definition 5 An optimisation problem $\mathcal{Q}$ is *polynomially-bounded* if there is a polynomial q such that for every instance I of $\mathcal{Q}$, $opt(I) \leq q(|I|)$ (we reiterate that in general the value $opt(I)$ might be exponential in $|I|$). We denote the class of polynomially-bounded P-optimisation problems by P_{opt}^{PB}, the (sub-)class of polynomially-bounded P-maximisation problems by P_{max}^{PB}, and the (sub-)class of polynomially-bounded P-minimisation problems by P_{min}^{PB}. There are analogous definitions of NP_{opt}^{PB}, NP_{max}^{PB}, and NP_{min}^{PB}. □

We now mention some examples of optimisation problems.

Example 6 Consider the maximum flow problem MAXUNITFLOW $= (\mathcal{I}, \mathcal{F}, f, opt)$, where:

- $\mathcal{I}$ is the set of triples (G, s, t), with G a digraph and s and t two distinct vertices of G, where s has in-degree 0 and t has out-degree 0, so that all edges are assumed to have unit capacity;
- $\mathcal{F}((G, s, t))$ is the set of all possible flows for (G, s, t);
- $f(I, J)$, for some instance I and for some feasible solution $J \in \mathcal{F}(I)$, is the size of the flow J.

It is well-known that the decision version of MAXUNITFLOW is in P (see, e.g., [2]); thus, MAXUNITFLOW $\in \mathsf{P}_{max}$. In fact, MAXUNITFLOW $\in \mathsf{P}_{max}^{PB}$.

Example 7 Consider the maximum 2-satisfiability problem MAX2SAT $= (\mathcal{I}, \mathcal{F}, f, opt)$, where:

- $\mathcal{I}$ is the set of conjunctive normal form formulae φ where every clause has 2 literals;
- $\mathcal{F}(\varphi)$ is the set of truth assignments on the Boolean variables involved in φ;
- $f(I, J)$, for some instance I and for some feasible solution $J \in \mathcal{F}(I)$, is the number of clauses of I made *true* under the truth assignment J.

It is well-known that the decision version of MAX2SAT is NP-complete (see, e.g., [5]). Hence, MAX2SAT is in NP_{max}^{PB} (and not in P_{max} unless $\mathsf{P} = \mathsf{NP}$). If we define MAXHORN2SAT just as was MAX2SAT except that all instances are in addition Horn formulae then thanks to a result in [10] where the decision version of MAXHORN2SAT was shown to be NP-complete, we obtain that MAXHORN2SAT is in NP_{max}^{PB} and unlikely to be in P_{max}^{PB}. However, if we define the problem MAX2SAT($\leq$2) by restricting instances of MAX2SAT so that every variable appears in at most 2 clauses then as the decision version of MAX2SAT($\leq$2) can be solved in linear time [19], MAX2SAT($\leq$2) is in P_{max}^{PB}.

Example 8 Consider the minimum shortest-path problem MINSP $= (\mathcal{I}, \mathcal{F}, f, opt)$, where:

- $\mathcal{I}$ is the set of triples (G, s, t), with G a digraph and s and t two distinct vertices of G;
- $\mathcal{F}((G, s, t))$ is the set of all possible paths in G from s to t;
- $f(I, J)$, for some instance I and for some feasible solution $J \in \mathcal{F}(I)$, is the length of the path J.

It is well-known that the decision version of MINSP is in P (see, e.g., [2]); thus, MINSP $\in \mathsf{P}^{PB}_{min}$.

Henceforth, we define the maximum or the minimum of the empty set as being $\perp$.

2.3 Using logic to classify NP-optimization problems

We begin with some basic definitions. For us, a *signature* σ is a finite tuple of relation symbols $R_1, R_2, \ldots, R_r$, where each R_i has arity a_i, and constant symbols $C_1, C_2, \ldots, C_c$. A *finite structure over* σ, or σ*-structure*, $\mathcal{A}$ of size n, where $n \geq 2$, consists of a *domain*, or *universe*, $\{0, 1, \ldots, n-1\}$ and a relation R_i of arity a_i (resp. constant C_j), for every relation symbol R_i (resp. constant symbol C_j) in σ (it causes no confusion that we do not differentiate between relations and relation symbols, and constants and constant symbols). We denote both the size and the domain of a structure $\mathcal{A}$ as $|\mathcal{A}|$ (again, this causes no confusion). Let σ and τ be signatures with no symbols in common. Suppose that $\mathcal{A}$ is a σ-structure and $\mathcal{B}$ is a τ-structure with $|\mathcal{A}| = |\mathcal{B}|$. The $\sigma \cup \tau$-structure $(\mathcal{A}, \mathcal{B})$ has domain that of $\mathcal{A}$ (and $\mathcal{B}$) with relations and constants corresponding to symbols from σ (resp. τ) inherited from $\mathcal{A}$ (resp. $\mathcal{B}$). If $\tau = \langle S_1, S_2, \ldots, S_t \rangle$, where each S_i is a relation symbol, then we sometimes denote $(\mathcal{A}, \mathcal{B})$ by $(\mathcal{A}, S_1, S_2, \ldots, S_t)$. A *problem* is an isomorphism-closed set of finite structures over some fixed signature; so, a problem refers to a decision problem (as opposed to an optimisation problem). Of particular interest to us is a *successor relation*; that is, a binary relation over some domain of size n where this relation is of the form

$$\{(a_0, a_1), (a_1, a_2), \ldots, (a_{n-2}, a_{n-1}) : \text{ all } a_i\text{'s are distinct}\}.$$

We assume that the reader is familiar with using first-order logic FO and second-order logic SO to define problems (we refer the reader to texts such as [3, 7, 9, 15]).

Henceforth, all instances of some optimisation problem are finite structures $\mathcal{A}$ over some fixed signature, σ say, and we say that such an optimisation problem is *over* σ. We make no assumptions as regards the feasible solutions of some instance although in practice they tend to be structures over some (fixed) signature. Such a framework fully captures all of the optimisation problems from the previous section, for if one formally considers instances of optimisation problems as strings of symbols (as one must if one is to classify the solution of optimisation problems using a device like a Turing machine) then strings over an alphabet consisting of m symbols, say, can be considered as finite structures over a signature consisting of m unary relation symbols and a binary relation symbol where the corresponding binary relation of any structure is a successor relation, detailing an ordering of the elements of the structure, and for every element in the domain of the structure, exactly one of the unary relations holds. However, adopting our logical framework allows us to consider optimisation problems via a more natural representation than strings; for example, an optimisation problem whose instances are digraphs is more naturally described using a binary relation encoding adjacency matrices than as a string denoting the concatenation of the rows of adjacency matrices.

In [13], Kolaitis and Thakur characterized the classes of polynomially-bounded NP-maximization problems NP^{PB}_{max} and polynomially-bounded NP-minimization problems NP^{PB}_{min}.

Theorem 9 (Kolaitis and Thakur [13]) Let $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ be a maximisation problem over σ. The following are equivalent.

1. $\mathcal{Q}$ is a polynomially-bounded NP-maximization problem, i.e., $\mathcal{Q} \in \mathsf{NP}^{PB}_{max}$.
2. There exists a signature τ, consisting solely of relation symbols and disjoint from σ, and a first-order formula $\varphi(\mathbf{x})$ over $\sigma \cup \tau$, where $\mathbf{x}$ is a k-tuple of variables, for some k, such that for every instance $\mathcal{A} \in \mathcal{I}$:

$$opt(\mathcal{A}) = \max_{\mathcal{B}}\{|\{\mathbf{u} \in |\mathcal{A}|^k : (\mathcal{A}, \mathcal{B}) \models \varphi(\mathbf{u})\}|\},$$

   where $\mathcal{B}$ ranges over all τ-structures of size $|\mathcal{A}|$.

Moreover, if one of the above conditions holds then the formula φ, above, can be taken to be a Π_2-formula. □

Theorem 10 (Kolaitis and Thakur [13]) Let $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ be a minimisation problem over σ. The following are equivalent.

1. $\mathcal{Q}$ is a polynomially-bounded NP-minimization problem, i.e., $\mathcal{Q} \in \mathsf{NP}^{PB}_{min}$.
2. There exists a signature τ, consisting solely of relation symbols and disjoint from σ, and a first-order formula $\varphi(\mathbf{x})$ over $\sigma \cup \tau$, where $\mathbf{x}$ is a k-tuple of variables, for some k, such that for every instance $\mathcal{A} \in \mathcal{I}$:

$$opt(\mathcal{A}) = \min_{\mathcal{B}}\{|\{\mathbf{u} \in |\mathcal{A}|^k : (\mathcal{A}, \mathcal{B}) \models \varphi(\mathbf{u})\}|\},$$

   where $\mathcal{B}$ ranges over all τ-structures of size $|\mathcal{A}|$.

Moreover, if one of the above conditions holds then the formula φ, above, can be taken to be a Σ_2-formula. □

Note that in Theorems 9 and 10 a τ-structure $\mathcal{B}$ (of size $|\mathcal{A}|$) need not correspond to a feasible solution of $\mathcal{A}$ as is defined in $\mathcal{Q}$.

The class of (logically-defined) maximisation problems defined in Theorem 9 is called MAX Π_2 and the class of minimisation problems defined in Theorem 10 is called MIN Σ_2, with the notation derived from the syntax of the defining first-order formula. By imposing suitable restrictions upon the formula φ in Theorems 9 and 10, one obtains classes such as MAX Π_i, MAX Σ_i, MIN Π_i, and MIN Σ_i, for $i \geq 0$. Obviously, NP^{PB}_{max} = MAX Π_2 = MAX Π_i and NP^{PB}_{min} = MIN Σ_2 = MIN Σ_i, for all $i \geq 2$. In fact, Kolaitis and Thakur also proved the following result.

Theorem 11 (Kolaitis and Thakur [13])

- MAX $\Sigma_0 \subset$ MAX $\Sigma_1 \subset$ MAX Π_1 = MAX $\Sigma_2 \subset$ MAX $\Pi_2 = \mathsf{NP}^{PB}_{max}$.
- MIN Σ_0 = MIN $\Sigma_1 \subset$ MIN Π_1 = MIN Σ_2 = MIN $\Pi_2 = \mathsf{NP}^{PB}_{min}$. □

Kolaitis and Thakur observed in [14] that many natural optimisation problems are such that a feasible solution to an instance is a finite set of relations satisfying a first-order sentence and that the objective function is the cardinality of one of these relations. Consequently, Kolaitis and Thakur went on in [14] to vary their logical framework slightly. They defined the following classes of optimisation problems. Note that for a relation X, we write $|X|$ to denote the number of tuples in X.

Definition 12 Let $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ be a maximisation problem over σ and let $i \geq 1$. The optimisation problem $\mathcal{Q}$ is in MAX FΠ_i if, and only if, there exists a Π_i (first-order) sentence φ over $\sigma \cup \tau$, where $\tau = \langle S_1, S_2, \ldots, S_t \rangle$ and where each S_j is a relation symbol not appearing in σ, with the property that for every instance $\mathcal{A}$ of $\mathcal{I}$:

$$opt(\mathcal{A}) = \max_{\mathcal{B}} \{|S_1| : (\mathcal{A}, \mathcal{B}) \models \varphi\},$$

where $\mathcal{B}$ ranges over all τ-structures of size $|\mathcal{A}|$. □

The classes MAX FΣ_i, MIN FΠ_i, and MIN FΣ_i, for $i \geq 1$, are defined analogously.

Kolaitis and Thakur [14] showed that these new classes of optimisation problems are closely related to the classes discussed earlier.

Theorem 13 (Kolaitis and Thakur [14])

$$\left.\begin{array}{l}\text{MAX } \Sigma_0 \\ \text{MAX F}\Sigma_1\end{array}\right\} \subset \text{MAX } \Sigma_1 \subset \text{MAX } \Pi_1 = \text{MAX F}\Pi_1 = \text{MAX } \Sigma_2$$
$$= \text{MAX F}\Sigma_2 \subset \text{MAX } \Pi_2 = \text{MAX F}\Pi_2 = \mathsf{NP}^{PB}_{max}.$$

$$\left.\begin{array}{c}\text{MIN } \Sigma_0 = \text{MIN } \Sigma_1 = \text{MIN F}\Pi_1 \\ \text{MIN F}\Sigma_1\end{array}\right\} \subset \text{MIN F}\Sigma_2 \subset \text{MIN F}\Pi_2 = \text{MIN } \Pi_1$$
$$= \text{MIN } \Sigma_2 = \text{MIN } \Pi_2 = \mathsf{NP}^{PB}_{min}. \quad \Box$$

The bracketing used in the statement of Theorem 13 is to denote that the classes are incomparable.

2.4 Manyem's framework for P-optimization problems

Inspired by the work of Kolaitis and Thakur, in [16] Manyem (and subsequently with Bueno in [1]) attempted to provide a suitable logical framework to characterize the classes of polynomially-bounded P-maximisation problems and polynomially-bounded P-minimisation problems. Whereas Kolaitis and Thakur's logical framework had been derived from Fagin's seminal characterization of NP as the class of problems definable in existential second-order logic [4], Bueno and Manyem tried to take Grädel's characterization of P [6] as the class of problems definable in a particular fragment of existential second-order logic as their inspiration. We shall now describe Grädel's result.

We say that a quantifier-free first-order formula over $\sigma \cup \tau$, where τ consists entirely of relation symbols and is disjoint from σ, is a *Horn formula over* (σ, τ) if it is a conjunction of clauses where each clause contains at most one positive atom involving a symbol from τ. The logic ∃SO-Horn is the fragment of existential second-order logic consisting of all formulae over some signature σ of the form:

$$\exists S_1 \exists S_2 \ldots \exists S_t \forall y_1 \forall y_2 \ldots \forall y_m \varphi,$$

where each S_i is a relation symbol not in the underlying signature σ, each y_j is a (first-order) variable, and φ is a Horn formula over (σ, τ), where $\tau = \langle S_1, S_2, \ldots, S_t \rangle$. We say that a logic $\mathcal{L}$ *describes*, or *captures*, a class of (decision) problems $\mathcal{C}$ in the presence of a *built-in successor relation*, or *on ordered structures*, if the following are equivalent:

- the problem Ω, over the signature σ, is in $\mathcal{C}$;
- there is a sentence Φ of $\mathcal{L}$ over the signature $\sigma \cup \langle succ, min, max \rangle$, where *succ* is a binary relation symbol not in σ and *min* and *max* are constant symbols not in σ, such that:
  - for every σ-structure $\mathcal{A}$, as to whether the expansion of $\mathcal{A}$ by a successor relation *succ* and constants *min* and *max*, so that *min* (resp. *max*) is the minimal (resp. maximal) element of the linear order described by *succ*, satisfies Φ is independent of the particular successor relation chosen (that is, Φ is *order invariant*);
  - for every σ-structure $\mathcal{A}$, $\mathcal{A} \in \Omega$ if, and only if, the expansion of $\mathcal{A}$ by some successor relation *succ* and corresponding constants *min* and *max* satisfies Φ.

For more on built-in successor relations, we refer the reader to [3, 7, 9, 15]. One bit of notational convenience we use is that when we talk of a sentence of some logic over some signature σ in the presence of a built-in successor relation, we suppress mention of the binary relation symbol *succ* and the two constant symbols *min* and *max* even though they are present and need to be instantiated with some successor relation and two constants in order for us to interpret the sentence in some σ-structure. Also, when we say that a (quantifier-free first-order) formula is a Horn formula over (σ, τ) then we impose no conditions as regards the occurrence of atoms involving the relation symbol *succ* and the constant symbols *min* and *max* (unless *min* or *max* appears in some atom involving a relation symbol from τ).

Theorem 14 (Grädel [6]) A problem is in P if, and only if, it can be defined by a sentence of ∃SO-Horn in the presence of a built-in successor relation. □

In [16], Manyem gave a definition of a P-optimisation problem and a logical framework within which to try and capture the classes of polynomially-bounded P-maximisation problems and polynomially-bounded P-minimisation problems. As we have already mentioned, his definition of a P-optimisation problem was at variance with the definition to be expected should one proceed analogously to related research on (NP) optimisation problems, mentioned above. In Theorem 3 of [16] he claims that every polynomially-bounded P-maximisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ over σ (according to his definition) is such that there exists a signature τ consisting of only relation symbols and disjoint from σ and a Horn formula $\varphi(\mathbf{y})$ over (σ, τ), where $\mathbf{y}$ is the tuple of free variables of φ, such that for every instance $\mathcal{A} \in \mathcal{I}$:

$$opt(\mathcal{A}) = \max_{\mathcal{B}} |\{\mathbf{u} : (\mathcal{A}, \mathcal{B}, \mathbf{u}) \models \forall x_1 \forall x_2 \ldots \forall x_k \varphi(\mathbf{y})\}|,$$

with $\mathcal{B}$ ranging over all τ-structures with domain $|\mathcal{A}|$ and $\mathbf{u}$ detailing values for the variables of $\mathbf{y}$. Manyem made a similar claim relating to polynomially-bounded P-minimisation problems in Theorem 10 of [16]. Manyem allowed for the use of a built-in successor relation in the formula φ, above, but did not explain how $\varphi(\mathbf{y})$ might be order-invariant; consequently, he left open the possibility that $|\{\mathbf{u} : (\mathcal{A}, \mathcal{B}, \mathbf{u}) \models \forall x_1 \forall x_2 \ldots \forall x_k \varphi(\mathbf{y})\}|$ might vary depending upon the particular successor relation chosen. Manyem made no claims as regards the converse direction; that is, whether optimisation problems definable in the above logical form are necessarily polynomially-bounded P-optimisation problems. Manyem attempted to demonstrate the efficacy of his framework by defining within it the problems MAXUNITFLOW and MINSP (see Examples 2.1.2 and 2.2.1 of [16])). Unfortunately there were errors in both definitions (with reference to [16]: the original formula in (25) is not equivalent to a universally quantified set of Horn clauses; nor is the formula φ_5 at the bottom of page 17). In the next section, we show how Manyem's framework cannot hope to be used to logically capture the class of polynomially-bounded P-maximisation problems or the class of polynomially-bounded P-minimization problems (in an analogous fashion to the characterizations of the classes of polynomially-bounded NP-maximisation and polynomially-bounded NP-minimisation problems as obtained by Kolaitis and Thakur).

3 The failure of Manyem's framework

We show how any framework defined in accordance with that proposed by Manyem will not suffice to characterize the classes of polynomially-bounded

P-maximisation problems and polynomially-bounded P-minimisation problems. We remind the reader of the maximisation problem MAXHORN2SAT of NP^{PB}_{max}: an instance of MAXHORN2SAT consists of a set of clauses where each clause is a Horn clause involving at most 2 literals; a feasible solution to such an instance is a truth assignment on the underlying Boolean variables; and the objective function is the number of clauses of the instance satisfied by the given truth assignment. The decision version of the maximisation problem MAXHORN2SAT was proven to be NP-complete in [10].

Theorem 15 There exists a polynomially-bounded maximisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ such that:

- $\mathcal{Q}$ is over $\sigma = \langle H, Z\rangle$, where H is a relation symbol of arity 4 and Z is a constant symbol, with $\mathcal{I}$ the set of σ-structures;
- $\tau = \langle P, T\rangle$, where P and T are both relation symbols of arity 1, so that the set of feasible solutions $\mathcal{F}(\mathcal{A})$ to some instance $\mathcal{A} \in \mathcal{I}$ is the set of τ-structures with domain $|\mathcal{A}|$;
- φ is a Horn formula over (σ, τ) with free variables x_1, x_2, x_3, y, so that given some instance $\mathcal{A} \in \mathcal{I}$ and some feasible solution $\mathcal{B} \in \mathcal{F}(\mathcal{I})$,

$$f(\mathcal{A}, \mathcal{B}) = |\{u : (\mathcal{A}, \mathcal{B}, u) \models \forall x_1 \forall x_2 \forall x_3 \varphi(y)\}|;$$

- the decision version of $\mathcal{Q}$ is NP-complete.

Proof Let I be an instance of the decision version of MAXHORN2SAT of size n; so, w.l.o.g. I involves clauses $C_1, C_2, \ldots, C_n$ and Boolean variables $X_1, X_2, \ldots, X_n$. Every clause is of one of the following forms:

(*i*) $X_i \Rightarrow X_j$

(*ii*) $X_i \wedge X_j \Rightarrow \mathit{false}$

(*iii*) $\mathit{true} \Rightarrow X_i$

(*iv*) $X_i \Rightarrow \mathit{false}$.

Define the signature $\sigma = \langle H, Z\rangle$, where H is a relation symbol of arity 4 and Z is a constant symbol, and the signature $\tau = \langle P, T\rangle$, where P and T are both relation symbols of arity 1. Let $\Phi(I)$ be the σ-structure with domain $\{0, 1, \ldots, n\}$, with the constant $Z = 0$ and with the relation H defined as follows:

(*i*) if clause C_k of I is of the form $X_i \Rightarrow X_j$ then $(i, 0, j, k) \in H$;

(*ii*) if clause C_k of I is of the form $X_i \wedge X_j \Rightarrow \mathit{false}$ then $(i, j, 0, k) \in H$;

(*iii*) if clause C_k of I is of the form $\mathit{true} \Rightarrow X_i$ then $(0, 0, i, k) \in H$;

(*iv*) if clause C_k of I is of the form $X_i \Rightarrow \mathit{false}$ then $(i, 0, 0, k) \in H$.

This completely defines H.

Define the formula Ψ' over $\sigma \cup \tau$ as

$$
\begin{aligned}
&\forall x_1 \forall x_2 \forall x_3 (\\
&\qquad ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, Z, x_3, x_1) \wedge P(x_2) \wedge \neg P(x_3)) \\
&\qquad\qquad \Rightarrow \neg T(x_1)) \\
&\wedge \quad ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, x_3, Z, x_1) \wedge P(x_2) \wedge P(x_3)) \\
&\qquad\qquad \Rightarrow \neg T(x_1)) \\
&\wedge \quad ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(Z, Z, x_2, x_1) \wedge \neg P(x_2)) \Rightarrow \neg T(x_1)) \\
&\wedge \quad ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, Z, Z, x_1) \wedge P(x_2)) \Rightarrow \neg T(x_1)) \\
&\wedge \quad \neg T(Z)).
\end{aligned}
$$

Note that the matrix of Ψ' is a Horn formula over (σ, τ).

We claim that there exists a truth assignment on $X_1, X_2, \ldots, X_n$ making at least m clauses of I *true* if, and only if, there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi'$ and $|T| \geq m$.

Suppose that π is a truth assignment on $X_1, X_2, \ldots, X_n$ making at least m clauses of I *true*. Define

$$P = \{i : 1 \leq i \leq n, \pi(X_i) = \textit{true}\}$$

and

$$T = \{k : 1 \leq k \leq n, \pi \text{ makes clause } C_k \text{ of } I \ \textit{true}\}.$$

There are 4 cases to consider.

(*i*) Suppose that C_k is of the form $X_i \Rightarrow X_j$ and that π makes C_k *false*; so, $\pi(X_i) = \textit{true}$ and $\pi(X_j) = \textit{false}$. Hence, $P(i)$ and $\neg P(j)$ hold. Consequently, $H(i, 0, j, k) \wedge P(i) \wedge \neg P(j)$ holds. Furthermore, $\neg T(k)$ holds and so

$$(H(i, 0, j, k) \wedge P(i) \wedge \neg P(j)) \Rightarrow \neg T(k) \text{ holds.}$$

If π makes C_k *true* then $\pi(X_i) = \textit{false}$ or $\pi(X_j) = \textit{true}$, and so at least one of $\neg P(i)$ and $P(j)$ holds with the consequence that

$$(H(i, 0, j, k) \wedge P(i) \wedge \neg P(j)) \Rightarrow \neg T(k) \text{ holds.}$$

(*ii*) Suppose that C_k is of the form $X_i \wedge X_j \Rightarrow \textit{false}$ and that π makes C_k *false*; so, $\pi(X_i) = \textit{true}$ and $\pi(X_j) = \textit{true}$. Hence, $P(i)$ and $P(j)$ hold. Consequently, $H(i, j, 0, k) \wedge P(i) \wedge P(j)$ holds. Furthermore, $\neg T(k)$ holds and so

$$(H(i, j, 0, k) \wedge P(i) \wedge P(j)) \Rightarrow \neg T(k) \text{ holds.}$$

If π makes C_k *true* then $\pi(X_i) = \textit{false}$ or $\pi(X_j) = \textit{false}$, and so at least one of $\neg P(i)$ and $\neg P(j)$ holds with the consequence that

$$(H(i, j, 0, k) \wedge P(i) \wedge P(j)) \Rightarrow \neg T(k) \text{ holds.}$$

(*iii*) Suppose that C_k is of the form *true* $\Rightarrow X_i$ and that π makes C_k *false*; so, $\pi(X_i) =$ *false*. Hence, $\neg P(i)$ holds. Consequently, $H(0,0,i,k) \wedge \neg P(i)$ holds. Furthermore, $\neg T(k)$ holds and so

$$(H(0,0,i,k) \wedge \neg P(i)) \Rightarrow \neg T(k) \text{ holds.}$$

If π makes C_k *true* then $\pi(X_i) =$ *true*, and so $P(i)$ holds with the consequence that

$$(H(0,0,i,k) \wedge \neg P(i)) \Rightarrow \neg T(k) \text{ holds.}$$

(*iv*) Suppose that C_k is of the form $X_i \Rightarrow$ *false* and that π makes C_k *false*; so, $\pi(X_i) =$ *true*. Hence, $P(i)$ holds. Consequently, $H(i,0,0,k) \wedge P(i)$ holds. Furthermore, $\neg T(k)$ holds and so

$$(H(i,0,0,k) \wedge P(i)) \Rightarrow \neg T(k) \text{ holds.}$$

If π makes C_k *true* then $\pi(X_i) =$ *false*, and so $\neg P(i)$ holds with the consequence that

$$(H(i,0,0,k) \wedge P(i)) \Rightarrow \neg T(k) \text{ holds.}$$

Consequently, $|T| \geq m$ and by definition of H, $(\Phi(I), P, T) \models \Psi'$.

Conversely, suppose that there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi'$ and $|T| \geq m$ (note that $\neg T(0)$ holds). Define the truth assignment π on $X_1, X_2, \ldots, X_n$ via

$$\pi(X_i) = \textit{true} \text{ if, and only if, } P(i) \text{ holds,}$$

for all $i = 1, 2, \ldots, n$. Consider the clause C_k of I. Again, there are 4 cases.

(*i*) Suppose that $T(k)$ holds and that C_k is of the form $X_i \Rightarrow X_j$; thus, $H(i,0,j,k)$ holds in $\Phi(I)$. As $(\Phi(I), P, T) \models \Psi'$, we must have that $\neg P(i) \vee P(j)$ holds. Thus, $\pi(X_i) =$ *false* or $\pi(X_j) =$ *true* and we have that clause C_k of I is *true* under π.

(*ii*) Suppose that $T(k)$ holds and that C_k is of the form $X_i \wedge X_j \Rightarrow$ *false*; thus, $H(i,j,0,k)$ holds in $\Phi(I)$. As $(\Phi(I), P, T) \models \Psi'$, we must have that $\neg P(i) \vee \neg P(j)$ holds. Thus, $\pi(X_i) =$ *false* or $\pi(X_j) =$ *false* and we have that clause C_k of I is *true* under π.

(*iii*) Suppose that $T(k)$ holds and that C_k is of the form *true* $\Rightarrow X_i$; thus, $H(0,0,i,k)$ holds in $\Phi(I)$. As $(\Phi(I), P, T) \models \Psi'$, we must have that $P(i)$ holds. Thus, $\pi(X_i) =$ *true* and we have that clause C_k of I is *true* under π.

(*iv*) Suppose that $T(k)$ holds and that C_k is of the form $X_i \Rightarrow$ *false*; thus, $H(i,0,0,k)$ holds in $\Phi(I)$. As $(\Phi(I), P, T) \models \Psi'$, we must have that $\neg P(i)$ holds. Thus, $\pi(X_i) =$ *false* and we have that clause C_k of I is *true* under π.

Consequently, the truth assignment π makes at least m clauses of I *true*.

Define Ψ as $\Psi' \wedge T(y)$, where y is a new variable. Define the maximisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ as follows:

- an instance of $\mathcal{I}$ is a σ-structure;
- a feasible solution to an instance $\mathcal{A} \in \mathcal{I}$ is a τ-structure whose domain is $|\mathcal{A}|$;
- the objective function $f(\mathcal{A}, \mathcal{B})$, where $\mathcal{A} \in \mathcal{I}$ and $\mathcal{B} \in \mathcal{F}(\mathcal{A})$, is given by $|\{u : (\mathcal{A}, \mathcal{B}, u) \models \Psi(y)\}|$.

We can clearly construct the instance $\Phi(I)$ of the decision version of $\mathcal{Q}$ from the instance I of the decision version of MAXHORN2SAT in polynomial-time. Hence, using the result from [10] that the decision version of MAXHORN2SAT is NP-complete, we obtain that the decision version of $\mathcal{Q}$ is NP-complete too, and our result follows. □

The minimisation problem MINHORN2SAT from [12] (where it was called 2-MINSAT) is defined exactly as was MAXHORN2SAT except that instead of trying to maximise the number of clauses satisfied, we try to minimise the number of clauses satisfied. It was proven in [12] that the decision version of MINHORN2SAT is NP-complete.

Theorem 16 There exists a polynomially-bounded minimisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ such that:

- $\mathcal{Q}$ is over $\sigma = \langle H, Z \rangle$, where H is a relation symbol of arity 4 and Z is a constant symbol, with $\mathcal{I}$ the set of σ-structures;
- $\tau = \langle P, T \rangle$, where P and T are both relation symbols of arity 1, so that the set of feasible solutions $\mathcal{F}(\mathcal{A})$ to some instance $\mathcal{A} \in \mathcal{I}$ is the set of τ-structures with domain $|\mathcal{A}|$;
- φ is a Horn formula over (σ, τ) with free variables x_1, x_2, x_3, y, so that given some instance $\mathcal{A} \in \mathcal{I}$ and some feasible solution $\mathcal{B} \in \mathcal{F}(\mathcal{I})$,

$$f(\mathcal{A}, \mathcal{B}) = |\{u : (\mathcal{A}, \mathcal{B}, u) \models \forall x_1 \forall x_2 \forall x_3 \varphi(y)\}|;$$

- the decision version of $\mathcal{Q}$ is NP-complete.

Proof Our proof is very similar to that of Theorem 15 and so we only highlight the essential differences. Adopting the nomenclature of the proof of Theorem 15, we proceed exactly as we did in that proof except that instead of defining Ψ'

we define the formula Ψ'' as

$$\begin{array}{ll} \multicolumn{2}{l}{\forall x_1 \forall x_2 \forall x_3 (} \\ & ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, Z, x_3, x_1) \wedge (\neg P(x_2) \vee P(x_3))) \\ & \hfill \Rightarrow \neg T(x_1)) \\ \wedge & ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, x_3, Z, x_1) \wedge (\neg P(x_2) \vee \neg P(x_3))) \\ & \hfill \Rightarrow \neg T(x_1)) \\ \wedge & ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(Z, Z, x_2, x_1) \wedge P(x_2)) \Rightarrow \neg T(x_1)) \\ \wedge & ((x_1 \neq Z \wedge x_2 \neq Z \wedge x_3 \neq Z \wedge H(x_2, Z, Z, x_1) \wedge \neg P(x_2)) \Rightarrow \neg T(x_1)) \\ \wedge & T(Z)). \end{array}$$

Note that the matrix of Ψ'' can easily be converted to an equivalent Horn formula over (σ, τ).

We claim that there exists a truth assignment on $X_1, X_2, \ldots, X_n$ making at most m clauses of I *true* if, and only if, there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi''$ and $|T| \geq n + 1 - m$.

Suppose that π is a truth assignment on $X_1, X_2, \ldots, X_n$ making at most m clauses of I *true*. Define

$$P = \{i : 1 \leq i \leq n, \pi(X_i) = \textit{true}\}$$

and

$$T = \{k : 1 \leq k \leq n, \pi \text{ makes clause } C_k \text{ of } I \textit{ false}\} \cup \{0\}.$$

Suppose that C_k is of the form $X_i \Rightarrow X_j$ and that π makes C_k *true*. So, $\pi(X_i) = \textit{false}$ or $\pi(X_j) = \textit{true}$, and $\neg P(i)$ or $P(j)$ holds. Thus, $H(i, 0, j, k) \wedge (\neg P(i) \vee P(j))$ holds. Also, $\neg T(k)$ holds and so

$$(H(i, 0, j, k) \wedge (\neg P(i) \vee P(j))) \Rightarrow \neg T(k) \text{ holds.}$$

If π makes C_k *false* then $\pi(X_i) = \textit{true}$ and $\pi(X_j) = \textit{false}$, and $P(i)$ and $\neg P(j)$ hold. Thus,

$$(H(i, 0, j, k) \wedge (\neg P(i) \vee P(j))) \Rightarrow \neg T(k) \text{ holds.}$$

The same reasoning applies in each of the other 3 cases (corresponding to the different forms of C_k). Hence, if there exists a truth assignment on $X_1, X_2, \ldots, X_n$ making at most m clauses of I *true* then there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi''$ and $|T| \geq n + 1 - m$.

Conversely, suppose that there exist relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi''$ and $|T| \geq n + 1 - m$. Define the truth assignment π on $X_1, X_2, \ldots, X_n$ via

$$\pi(X_i) = \textit{true} \text{ if, and only if, } P(i) \text{ holds.}$$

Suppose that $T(k)$ holds and that C_k is of the form $X_i \Rightarrow X_j$; so, $H(i, 0, j, k)$ holds in $\Phi(I)$. As $(\Phi(I), P, T) \models \Psi''$, we must have that $P(i) \wedge \neg P(j)$ holds.

Thus, $\pi(X_i) = \mathit{true}$ and $\pi(X_j) = \mathit{false}$, with π making C_k *false*. The same reasoning applies in each of the other 3 cases. Hence, if there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi''$ and $|T| \geq n+1-m$ then there exists a truth assignment on $X_1, X_2, \ldots, X_n$ making at most m clauses of I *true* (note that $T(Z)$ holds). Thus, there exists a truth assignment on $X_1, X_2, \ldots, X_n$ making at most m clauses of I *true* if, and only if, there exist unary relations P and T over $\{0, 1, \ldots, n\}$ such that $(\Phi(I), P, T) \models \Psi''$ and $|\neg T| \leq m$, where $\neg T = \{u : \neg T(u) \text{ holds}\}$.

By defining Ψ as $\Psi'' \wedge \neg T(y)$ and the minimisation problem $\mathcal{Q}$ just as we defined the maximisation problem $\mathcal{Q}$ in the proof of Theorem 15, reasoning as in the proof of Theorem 15, except using the NP-completeness of the decision version of the problem MINHORN2SAT, yields the result. □

An immediate consequence of Theorems 15 and 16 is that any framework based around Grädel's characterisation of P using restricted (Horn) formulae of existential second-order logic (as advocated by Manyem) will not characterize the class of polynomially-bounded P-maximisation problems nor the class of polynomially-bounded P-minimisation problems (assuming that $\mathsf{P} \neq \mathsf{NP}$).

Note also that when working with polynomially-bounded P-minimisation problems, there is a pronounced difference between the original framework proposed by Kolaitis and Thakur [13], where in order to obtain the objective function value we count the number of elements satisfying some formula, and the amended one [14], where we count the cardinality of a witnessing relation (Manyem chose to adopt the former framework when he strove for a logical classification of P-optimisation problems in [16]). Whilst MIN $\mathrm{F}\Pi_1 \subset \mathrm{MIN}\Pi_1$, with MIN $\mathrm{F}\Pi_1$ still containing NP-hard optimisation problems (like VERTEX COVER), if we have some optimisation problem $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, \mathit{opt})$, over σ, where for every instance $\mathcal{A} \in \mathcal{I}$ for which $\mathcal{F}(\mathcal{A})$ is non-empty,

$$\mathit{opt}(\mathcal{A}) = \max_{\mathcal{B}}\{|B_0| : (\mathcal{A}, \mathcal{B}) \models \forall x_1 \forall x_2 \ldots \forall x_k \varphi\},$$

with φ a Horn sentence over (σ, τ), $\mathcal{B}$ ranging over all τ-structures with domain $|\mathcal{A}|$ and B_0 a specific relation from $\mathcal{B}$, then $\mathcal{Q}$ is indeed a polynomially-bounded P-optimisation problem. To see this, simply 'expand' the sentence φ in any instance $\mathcal{A}$ in order to obtain a collection of Horn formulae. If there exists a witnessing set of relations $\mathcal{B}$ then there is a unique 'minimal' set of relations $\mathcal{B}$, computable in polynomial-time. The cardinality of the relation B_0 from this set of relations $\mathcal{B}$ yields the value $\mathit{opt}(\mathcal{A})$. Thus, if we were to adopt the amended framework from [14] and Manyem's approach to obtaining a logical characterization of polynomially-bounded P-minimisation problems then it is feasible that we might be able to do so (though we would have to ensure that all defining formulae were order-invariant, as we explained at the end of Section 2.4). Even this revised framework would fail for polynomially-bounded P-maximisation problems, though, as is demonstrated by the proof of Theorem 15 (as always, we assume that $\mathsf{P} \neq \mathsf{NP}$).

4 Logically capturing P^{PB}_{max} and P^{PB}_{min}

In this section we provide logical characterizations of the classes of polynomially-bounded P-maximisation problems and polynomially-bounded P-minimisation problems. The logic we use is not a fragment of existential second-order logic but the well-known inflationary fixed-point logic FO(IFP). We use the inductive process of building fixed-points in order to provide values for the objective functions of our optimisation problems.

4.1 Inflationary fixed-point logic

Let us begin by reminding the reader as to the definition of the logic FO(IFP) (we only provide brief definitions here and refer the reader to any of [3, 7, 9, 15] for more substantive details).

Let σ be some signature and let R be a k-ary relation symbol not in σ. Suppose that $\varphi(R, \mathbf{x})$ is some formula of some logic over $\sigma \cup \langle R \rangle$ whose free variables are those of the k-tuple $\mathbf{x}$. Let $\mathcal{A}$ be some σ-structure. We build the inflationary fixed-point of $\varphi(R, \mathbf{x})$ in $\mathcal{A}$ as follows. We define (the k-ary relation) R^0 over $|\mathcal{A}|$ as being the empty set. For $i \geq 1$, we define

$$R^i = \{\mathbf{u} \in |\mathcal{A}|^k : R^{i-1}(\mathbf{u}) \text{ holds or } \varphi(R^{i-1}, \mathbf{u}) \text{ holds in } \mathcal{A}\}.$$

The *inflationary fixed-point of* $\varphi(R, \mathbf{x})$ *in* $\mathcal{A}$ is defined as R^i, where i is the least integer for which $R^i = R^{i+1}$, and the *inductive depth* of this inflationary fixed-point is i.

Inflationary fixed-point logic FO(LFP) is built using the usual first-order constructs as well as the IFP operator, which allows us to construct inflationary fixed-points. The IFP operator is applied as follows. Suppose that φ is a formula of FO(IFP) over σ that involves an additional k-ary relation symbol R (not in σ) and has free variables those of the k-tuple $\mathbf{x}$ and those of the m-tuple $\mathbf{y}$. The formula $[\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{z})$ is a formula of FO(IFP), where $\mathbf{z}$ is a tuple of variables and constant symbols, and the free variables are those of the tuple $\mathbf{y}$ together with any variables appearing in $\mathbf{z}$ (in particular, the occurrences of the variables of $\mathbf{x}$ in φ are bound by the application of IFP). Note that φ might simply be a first-order formula or it might already involve applications of the IFP operator.

Let $\mathcal{A}$ be a σ-structure. Assuming that $\mathbf{u}$ is a tuple of elements of $|\mathcal{A}|$ providing values for the free variables of $[\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{z})$, we say that $[\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{z})$ holds in $(\mathcal{A}, \mathbf{u})$ if (the interpretation of) $\mathbf{z}$ lies in the inflationary fixed-point of $\varphi(R, \mathbf{x})$ in $(\mathcal{A}, \mathbf{u})$. We write $depth_{(\mathcal{A},\mathbf{u})}([\text{IFP}_{R,\mathbf{x}}\varphi])$ to denote the inductive depth of the inflationary fixed-point of $\varphi(R, \mathbf{x})$ in $(\mathcal{A}, \mathbf{u})$.

Inflationary fixed-point logic has been extensively studied within finite model theory. Pertinent to this paper is the following result where we denote the logic FO(IFP) in the presence of a built-in successor relation by FO_s(IFP).

Theorem 17 (Immerman [8], Vardi [20]) A problem is in P if, and only if, it can be defined by a sentence of FO_s(IFP). Moreover, any sentence of FO_s(IFP) is logically equivalent to one of the form $[\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{max})$, where φ is quantifier-free first-order and **max** is a tuple every component of which is the constant symbol max.

4.2 Our logical characterisations

We now use the inflationary fixed-point logic $\text{FO}_s(\text{IFP})$ to provide logical characterisations of the classes of polynomially-bounded P-maximisation problems P^{PB}_{max} and polynomially-bounded P-minimisation problems P^{PB}_{min}. However, before we do so we need another definition relating to formulae of $\text{FO}_s(\text{IFP})$. Let φ be a formula of $\text{FO}_s(\text{IFP})$ over the signature $\sigma \cup \langle R \rangle$, where R is a k-ary relation symbol, so that the free variables of φ are those of the k-tuple of variables $\mathbf{x}$. We say that $\varphi(R, \mathbf{x})$ is *depth-invariant* if the inductive depth of the inflationary fixed-point of $\varphi(R, \mathbf{x})$ in any σ-structure is independent of the actual underlying successor relation.

Theorem 18 Let $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ be a maximisation problem over σ. The following are equivalent.

1. $\mathcal{Q}$ is a polynomially-bounded P-maximisation problem (i.e., $\mathcal{Q} \in \mathsf{P}^{PB}_{max}$).
2. There exists some depth-invariant formula $\varphi(R, \mathbf{x})$ of $\text{FO}_s(\text{IFP})$ over $\sigma \cup \langle R \rangle$, where R is a k-ary relation symbol and the free variables of φ are those of the k-tuple $\mathbf{x}$, such that for any $\mathcal{A} \in \mathcal{I}$:
   - if $\mathcal{F}(\mathcal{A})$ is non-empty then $\mathcal{A} \models [\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{min})$ and the optimal value $\text{opt}(\mathcal{A})$ is given by $depth_{\mathcal{A}}([\text{IFP}_{R,\mathbf{x}}\varphi]) - 1$;
   - if $\mathcal{F}(\mathcal{A})$ is empty then $\mathcal{A} \not\models [\text{IFP}_{R,\mathbf{x}}\varphi](\mathbf{min})$.

Proof Suppose that $\mathcal{Q} \in \mathsf{P}^{PB}_{max}$ and let k be such that $\text{opt}(\mathcal{A}) < |\mathcal{A}|^k$, for all structures $\mathcal{A} \in \mathcal{I}$ for which $\mathcal{F}(\mathcal{A})$ is non-empty. Let R be a new k-ary relation symbol (that is not in σ). By definition, the decision problem $\mathcal{Q}'$ over $\sigma \cup \langle R \rangle$ defined as

$$\begin{aligned}\{(\mathcal{A}, R) \quad : \quad & \mathcal{A} \in \mathcal{I}, R \subseteq |\mathcal{A}|^k \text{ and there is a feasible solution} \\ & \mathcal{B} \in \mathcal{F}(\mathcal{A}) \text{ s.t. } f(\mathcal{A}, \mathcal{B}) \;\geq\; |R|\}\end{aligned}$$

is in P. By Theorem 17, there exists some sentence $[\text{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max})$ of the logic $\text{FO}_s(\text{IFP})$ for which

$$(\mathcal{A}, R) \in \mathcal{Q}' \text{ if, and only if, } (\mathcal{A}, R) \models [\text{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max}),$$

where $S \notin \sigma$ is some m-ary relation symbol (and hence $\mathbf{y}$ is an m-tuple of variables), and where ψ is quantifier-free first-order over $\sigma \cup \langle R, S \rangle$ and has free variables those of the tuple $\mathbf{y}$ (note that it is not necessarily the case that $m = k$).

Now, define the formula $\varphi_{max}(R, \mathbf{x})$ as:

$$[\text{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max}) \wedge (\mathbf{x} = \mathbf{min} \vee \exists \mathbf{z}(\mathbf{x} = \mathbf{z} + 1 \wedge R(\mathbf{z})))$$

(we write: $\mathbf{x} = \mathbf{z}+1$, for example, as short-hand for the quantifier-free first-order formula that expresses that $\mathbf{x}$ is the successor of $\mathbf{z}$ in the lexicographic order on k-tuples induced by the built-in successor relation; $\mathbf{x} = \mathbf{min}$, for example, as short-hand for $x_1 = min \wedge x_2 = min \wedge \ldots \wedge x_k = min$; and $\exists \mathbf{z}$, for example, as short-hand for $\exists z_1 \exists z_2 \ldots \exists z_k$). The formula $\varphi_{max}(R, \mathbf{x})$ is clearly depth-invariant. We claim that for any σ-structure $\mathcal{A} \in \mathcal{I}$ for which $\mathcal{F}(\mathcal{A})$ is non-empty,

$$\mathrm{opt}(\mathcal{A}) = depth_{\mathcal{A}}([\mathrm{IFP}_{R,\mathbf{x}}\varphi_{max}]) - 1.$$

Suppose that $\mathcal{A} \in \mathcal{I}$ with the set of feasible solutions $\mathcal{F}(\mathcal{I})$ non-empty; consequently, if R^0 is the empty relation then $(\mathcal{A}, R^0) \in \mathcal{Q}'$; that is, $(\mathcal{A}, R^0) \models [\mathrm{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max})$. Suppose that R^i is the relation constructed after the ith iteration of the IFP operator and that:

- $(\mathcal{A}, R^i) \models [\mathrm{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max})$
- R^i consists of the i smallest k-tuples from $|\mathcal{A}|^k$ (w.r.t. the lexicographic ordering induced by the built-in successor relation).

Thus, R^{i+1} consists of R^i plus the $(i+1)$th smallest k-tuple from $|\mathcal{A}|^k$. By induction, if R^i is the inflationary fixed-point of $\varphi_{max}(R, \mathbf{x})$ in $\mathcal{A}$ then: $|R^i| = i$ and $i = depth_{\mathcal{A}}([\mathrm{IFP}_{R,\mathbf{x}}\varphi_{max}]$; and $opt(\mathcal{A}) = i - 1$. Also, $\mathcal{A} \models [\mathrm{IFP}_{R,\mathbf{x}}\varphi_{max}](\mathbf{min})$. Alternatively, suppose that the set of feasible solutions $\mathcal{F}(\mathcal{I})$ is empty. So, if R^0 is the empty relation then $(\mathcal{A}, R^0) \not\models [\mathrm{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max})$ and $\mathcal{A} \not\models [\mathrm{IFP}_{R,\mathbf{x}}\varphi_{max}](\mathbf{min})$.

Conversely, suppose that $\mathcal{Q}$ satisfies the hypothesis in 2. in the statement of the theorem. As $opt(\mathcal{A})$ is bounded above by $|\mathcal{A}|^k$, for any instance $\mathcal{A} \in \mathcal{I}$, in order to show that $\mathcal{Q}$ is in P^{PB}_{max} we simply need to show that the decision problem consisting of deciding whether for any given instance $\mathcal{A} \in \mathcal{I}$ and any given integer m, $opt(\mathcal{A}) \geq m$, is solvable in polynomial-time. However, as the canonical algorithm computing the inflationary fixed-point of the formula $[\mathrm{IFP}_{R,\mathbf{x}}\varphi]$ in some instance $\mathcal{A} \in \mathcal{I}$ runs in time polynomial in $|\mathcal{A}|$, we can easily augment this algorithm with a clock in order to count the number of iterative steps, and thus the inductive depth of this inflationary fixed-point, in polynomial-time too. Thus, $\mathcal{Q} \in \mathsf{P}^{PB}_{max}$ as required. □

Theorem 19 Let $\mathcal{Q} = (\mathcal{I}, \mathcal{F}, f, opt)$ be a minimisation problem over σ. The following are equivalent.

1. $\mathcal{Q}$ is a polynomially-bounded P-minimisation problem (i.e., $\mathcal{Q} \in \mathsf{P}^{PB}_{min}$).
2. There exists some depth-invariant formula $\varphi(R, \mathbf{x})$ of $\mathrm{FO}_s(\mathrm{IFP})$ over $\sigma \cup \langle R \rangle$, where R is a k-ary relation symbol and the free variables of φ are those of the k-tuple $\mathbf{x}$, such that for any $\mathcal{A} \in \mathcal{I}$:
   - if $\mathcal{F}(\mathcal{A})$ is non-empty then $\mathcal{A} \models [\mathrm{IFP}_{R,\mathbf{x}}\varphi](\mathbf{min})$ and the optimal value $\mathrm{opt}(\mathcal{A})$ is given by $|\mathcal{A}|^k - depth_{\mathcal{A}}([\mathrm{IFP}_{R,\mathbf{x}}\varphi]) + 1$;
   - if $\mathcal{F}(\mathcal{A})$ is empty then $\mathcal{A} \not\models [\mathrm{IFP}_{R,\mathbf{x}}\varphi](\mathbf{min})$.

Proof The proof is similar to that of Theorem 18 and so we only include brief details. We adopt the nomenclature of the proof of Theorem 18 throughout.

Suppose that $\mathcal{Q} \in \mathsf{P}^{PB}_{min}$ and let k be such that $\text{opt}(\mathcal{A}) < |\mathcal{A}|^k$, for all structures $\mathcal{A} \in \mathcal{I}$ for which $\mathcal{F}(\mathcal{A})$ is non-empty. Let R be a new k-ary relation symbol. Define the decision problem $\mathcal{Q}'$ over $\sigma \cup \langle R \rangle$ as

$$\{(\mathcal{A}, R) \;:\; \mathcal{A} \in \mathcal{I}, R \subseteq |\mathcal{A}|^k \text{ and there is a feasible solution } \mathcal{B} \in \mathcal{F}(\mathcal{A}) \text{ s.t. } f(\mathcal{A}, \mathcal{B}) \leq |R|\}.$$

As $\mathcal{Q}' \in \mathsf{P}$, by Theorem 17 there exists some sentence $[\text{IFP}_{S,\mathbf{y}}\psi'(R, S, \mathbf{y})](\mathbf{max})$ of the logic $\text{FO}_s(\text{IFP})$ for which

$$(\mathcal{A}, R) \in \mathcal{Q}' \text{ if, and only if, } (\mathcal{A}, R) \models [\text{IFP}_{S,\mathbf{y}}\psi'(R, S, \mathbf{y})](\mathbf{max}),$$

where $S \notin \sigma$ is some m-ary relation symbol, $\mathbf{y}$ is some m-tuple of variables, and ψ' is quantifier-free first-order over $\sigma \cup \langle R, S \rangle$ with free variables those of the tuple $\mathbf{y}$.

Define the formula $\varphi_{min}(R, \mathbf{x})$ as:

$$[\text{IFP}_{S,\mathbf{y}}\psi(R, S, \mathbf{y})](\mathbf{max}) \wedge (\mathbf{x} = \mathbf{min} \vee \exists \mathbf{z}(\mathbf{x} = \mathbf{z} + 1 \wedge R(\mathbf{z}))),$$

where ψ is ψ' with every occurrence of an atom involving R replaced with its negation. By reasoning as in the proof of Theorem 18, for any σ-structure $\mathcal{A} \in \mathcal{I}$ for which $\mathcal{F}(\mathcal{A})$ is non-empty, we have that

$$\text{opt}(\mathcal{A}) = |\mathcal{A}|^k - \mathit{depth}_{\mathcal{A}}([\text{IFP}_{R,\mathbf{x}}\varphi_{min}]) + 1$$

and the result follows. □

Note that the actual numeric formulae giving the value of an optimal solution in Theorems 18 and 19 in terms of the depth of a fixed-point construction are to some extent unimportant. All that matters is that they are efficiently computable, which both formulae are. In consequence, we obtain logical characterizations of the classes P^{PB}_{max} and P^{PB}_{min}.

5 Conclusions

In this paper we have clarified the applicability of logical frameworks in relation to capturing classes of polynomially-bounded NP-optimisation problems. We have seen: that Manyem's framework does not (and will not) suffice; that there are additional differences between the two frameworks proposed by Kolaitis and Thakur when one restricts so as to consider P-optimisation problems; and that there does exist an alternative logical framework capturing polynomially-bounded P-optimisation problems.

We suggest the following as directions for further research. Whilst Manyem's attempt to capture classes of polynomial-time optimisation problems using fragments of existential second-order logic with the first-order quantifier-free part restricted to be a conjunction of Horn clauses has gone awry, it would be interesting to continue this investigation in relation to the hierarchy results from Theorem 13. That is, what happens at the lower end of this hierarchy when we restrict the first-order quantifier-free part of formulae to be a conjunction of Horn clauses, or even Krom clauses (a Krom clause is a clause with exactly 2 literals)? In order to work with the full class of P-optimisation problems, we need some way of sensibly incorporating the built-in successor relation into any framework. We surmise that this should be possible but will require some technical consideration. Of course, there is always the question of the relationships between restricted logically defined classes in the absence of any built-in relations.

Kolaitis and Thakur have shown that there are optimisation problems in MIN Π_1 that are not in MIN $\mathrm{F}\Pi_1$, and that both classes contain NP-hard minimisation problems. We have shown that restricting MIN $\mathrm{F}\Pi_1$ by enforcing that the quantifier-free part of formulae should be a conjugation of Horn clauses yields only P-minimisation problems, whereas doing likewise with MIN Π_1 can yield NP-hard minimisation problems. It would be interesting to better understand the relationship between the restricted versions of MIN Π_1 and MIN $\mathrm{F}\Pi_1$ (and between the restricted versions of MAX Π_1 and MAX $\mathrm{F}\Pi_1$, for that matter).

Finally, there is no doubt that polynomial-time optimisation problems are not as abundant as NP-optimisation problems, nor do they straddle the P versus NP divide as do NP-optimisation problems. Nevertheless, a more wide-ranging investigation as to the relationship between, for example, P-optimisation problems and optimisation problems that can be solved in NC or NL and into alternative means of logically defining P-optimisation problems is warranted.

Acknowledgement The authors are indebted to Prabhu Manyem for many clarifying conversations as regards his work, undertaken whilst he spent a sabbatical stay in Durham.

References

[1] Orestes Bueno and Prabhu Manyem. Polynomial-time maximisation classes: syntactic hierarchy. *Fundamenta Informaticae*, 84(1):111–133, 2008.

[2] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[3] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Monographs in Mathematics. Springer, 1999.

[4] Ronald Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity and Computation*, volume 7 of *SIAM-AMS Proceedings*, pages 43–73, 1974.

[5] Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.

[6] Erich Grädel. Capturing complexity classes by fragments of second-order logic. *Theoretical Computer Science*, 101(1):35–57, 1992.

[7] Erich Grädel, Phokion G. Kolaitis, Leonid Libkin, Martin Marx, Joel Spencer, Moshe Y. Vardi, Yde Venema, and Scott Weinstein. *Finite Model Theory and Its Applications.* Texts in Theoretical Computer Science. Springer, 2007.

[8] Neil Immerman. Relational queries computable in polynomial time. *Information and Control*, 68(1-3):86–104, 1986.

[9] Neil Immerman. *Descriptive Complexity.* Graduate Texts in Computer Science. Springer, 1999.

[10] Brigitte Jaumard and Bruno Simeone. On the complexity of the maximum satisfiability problem for horn formulas. *Information Processing Letters*, 26(1):1–4, 1987.

[11] David Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256–278, 1974.

[12] Rajeev Kohli, Ramesh Krishnamurti, and Prakash Mirchandani. The minimum satisfiability problem. *SIAM Journal of Discrete Mathematics*, 7(2):275–283, 1994.

[13] Phokion G. Kolaitis and Madhukar N. Thakur. Logical definability of NP optimization problems. *Information and Computation*, 115(2):321–353, 1994.

[14] Phokion G. Kolaitis and Madhukar N. Thakur. Approximation properties of NP minimization classes. *Journal of Computer and System Sciences*, 50(3):391–411, 1995.

[15] Leonid Libkin. *Elements of Finite Model Theory.* Texts in Theoretical Computer Science. Springer, 2004.

[16] Prabhu Manyem. Syntactic characterizations of polynomial time optimization classes. *Chicago Journal of Theoretical Computer Science*, 2008(3):1–23, May 2008.

[17] Alessandro Panconesi and Desh Ranjan. Quantifiers and approximation. *Theoretical Computer Science*, 107(1):145–163, 1993.

[18] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[19] Venkatesh Raman, Bala Ravikumar, and S. Srinivasa Rao. A simplified NP-complete MAXSAT problem. *Information Processing Letters*, 65(1):163–168, 1998.

[20] Moshe Y. Vardi. The complexity of relational query languages. In *Proceedings of 14th ACM Ann. Symp. on the Theory of Computing*, pages 137–146, 1982.