On the computational complexities of Quantified Integer Programming variants

Piotr Wojciechowski*
LDCSEE
West Virginia University
Morgantown, WV
pwojciec@mix.wvu.edu

K. Subramani *
LDCSEE
West Virginia University
Morgantown, WV
ksmani@csee.wvu.edu

Pavlos Eirinakis
DMST
Ath. Univ. of Econ. and Bus.
Athens, Greece
peir@aueb.gr

Abstract

In this paper, we study Quantified Integer Programming (QIP) and Quantified Integer Implication (QII) from the perspective of computational complexity. In particular, we show that the restricted language of QIIs is expressive enough to fully capture Presburger Arithmetic. Secondly, we establish the computational complexity of QIP over polytopes, i.e., cases where an upper and lower bound can always be derived for existential variables. Thirdly, we investigate the complexities of partially bounded and unbounded variants of QIPs and QIIs. Finally, we examine the connections between QIPs and QIIs with limited quantifier alternations and the polynomial hierarchy.

1 Introduction

This paper establishes the computational complexities of a number of mathematical programming models for uncertainty handling. In particular, it examines Quantified Integer Programming (QIP) and Quantified Integer Implication (QII). The abbreviations QIP and QII are used interchangeably to denote the class of problems or a specific instance, depending on the context. Both QIP and QII are versatile tools arising in a number of application domains including, but not limited to, real-time scheduling (Gerber et al., 1995) and program verification (Bradley and Manna, 2007). These paradigms generalize Integer Programming (IP) and Integer Implication in precisely the same manner as quantified satisfiability and quantified implication extend boolean satisfiability and boolean implication respectively.

The decision problem of QIPs and QIIs (and their variants) can be viewed as a 2-person **combinatorial**, **partizan** game (Ferguson, 2013). In such a game there are two players, viz., the existential player **X**, who chooses values for the existentially quantified integer variables, and the universal player **Y**, who chooses values for the universally quantified integer variables. **X** and **Y** make their choices according to the order of the variables in a quantifier string. Each choice corresponds to a move and each player has a predetermined number of moves. If, at the end of all the moves, the instantiated linear system in the QIP or the implication in the QII is true, then **X** wins the game (we say that **X** has a winning strategy). Otherwise, **Y** wins the game.

Moreover, this paper introduces the partially bounded and unbounded versions of QIP and QII. In the former, universally quantified variables are bounded below but not above. In the latter, universally quantified variables are not bounded at all. It is worth noting that the current work in QIPs and QIIs generalizes our previous work in Quantified Linear Programming (QLP) and Quantified Linear Implication (QLI) (Subramani, 2007; Eirinakis et al., 2014; Wojciechowski et al., 2015).

We show that several of these problems are complex enough to capture the entirety of Presburger Arithmetic (**PA**) (see Appendix A). A problem which can express Presburger Arithmetic is referred to as **PA-hard**. If, in addition, the problem is itself expressible in **PA**, then the problem is said to be **PA-complete**.

The principal contributions of this paper are as follows:

1. Establishing that QIPs over polytopes are **PSPACE-complete**.
2. Establishing that QII is **PA-complete**.
3. Establishing that the partially bounded and the unbounded variants of QIP and QII are all **PA-complete**.
4. Examining the relation of variants of QIP and QII with the **PH**.

The rest of the paper is organized as follows. The motivation for our work and related approaches in the literature are discussed in Section 2. In Section 3, we establish the computational complexity of QIP over polytopes. Section 4 and Section 5 are concerned with the computational complexities of the unbounded and partially bounded variants of QIP and QII respectively. Section 4 also includes a proof that QII is **PA-complete**. The relationship between QIP and the polynomial hierarchy (**PH**) is discussed in Section 6, while the relationship between QII and the **PH** is detailed in Section 7. We conclude in Section 8 by summarizing our contributions and identifying avenues for future research.

2 Motivation and Related Work

Modeling uncertainty is one of the principal application areas of QIP and QII. Many application models incorporate the assumption of constancy in data which is neither realistic

*This research was supported in part by the National Science Foundation through Award CCF-1305054.

nor accurate. In scheduling problems, for instance, the execution time of a job is usually considered fixed and known in advance. This simplifying assumption leads to simple tractable models. However, in real-time systems, such an assumption may lead to dire consequences (Stankovic et al., 1998).

In contrast, QIP and QII are ideal for expressing schedulability specifications. In real-time scheduling, a dispatcher wants to schedule a set of ordered, non-preemptive jobs within given time frames. The start time and the execution time of each job may vary among integer values. Hence, QIP can be utilized to model the start times as existentially quantified variables and the execution times as universally quantified variables, while linear constraints can be used to express the constraints between the various jobs and their start and end times (e.g., see (Subramani, 2003)). Thus, the existence of a feasible schedule is equivalent to deciding the corresponding QIP.

Consider the situation where a schedule has already been determined, but some constraints get modified. At this juncture, we can either compute a new schedule or check if the new constraint set is a relaxation of the initial constraint set. The latter approach is easily modeled as a QII with the initial constraint set acting as the antecedent and the new constraint set acting as the consequent of the implication.

Other approaches to scheduling are based on parametric flows (Serafini, 1996; McCormick, 2000) and selective assembly (Iwata et al., 1998). Such flow problems are naturally expressed as QIPs whose constraint matrix is totally unimodular.

In his seminal thesis, Tarski (1948) proved via quantifier elimination that the full elementary theory of real closed fields with addition, multiplication, and order is decidable. Since then, several other decision procedures have been proposed (e.g., see (Weispfenning, 1988; Collins and Hong, 1991; Dolzmann and Sturm, 1997; Dolzmann et al., 1998; Brown, 2003; Ratschan, 2006)). Moreover, several sub-classes of this theory have been studied. A quantifier elimination procedure for sentences in the theory of reals with addition and order exists, which is singly exponential in space and doubly exponential in time (Ferrante and Rackoff, 1975).

With respect to integer arithmetic, Peano arithmetic, i.e., the first-order theory of the natural numbers with addition, multiplication, and order, is not decidable. On the other hand, Presburger Arithmetic, which does not include multiplication, is decidable (Papadimitriou, 1994). In fact, the problem of deciding **PA** is super-exponential (Fischer and Rabin, 1974). **PA** is also studied in (Berman, 1980) (together with the theory of reals with addition and order), where an exponential time lower bound is provided and the time and space complexities of both theories at various levels of quantifier alternations are determined. QIP can be considered as a restriction to **PA**, permitting only conjunctions of linear inequalities. More elaborate relationships between linear inequalities can be expressed via QII.

The expressiveness of full first-order constraints is analyzed in (Colmerauer, 2003), where equality is considered as a unique relational symbol. Universally quantified interval constraints are examined in (Benhamou and Goulard, 2000) where their relationship with the computation of inner-approximations of real relations is established. The Fourier-Motzkin elimination technique for linear constraints over both rational domains and lattice domains is empirically studied in (Lassez and Maher, 1992). Polyhedral projection is further examined in (Lassez and Lassez, 1993) where the construction of the approximate convex hull of the feasible space permits an efficient decision procedure for certain clausal queries.

The Quantified Linear Programing paradigm was introduced in (Subramani, 2007). It has been shown that QLP is in **PSPACE**, although the hardness of this problem has not yet been established. Quantified Linear Implication with an arbitrary number of quantifier alternation has been shown to be **PSPACE-hard** (Eirinakis et al., 2014). It is not known whether this problem is in **PSPACE**. The computational complexities of several sub-classes of QLI for a given number of quantifier alternations have also been established. More specifically, QLIs are studied in (Eirinakis et al., 2012) in the context of entailment of parameterized linear constraints. In (Eirinakis et al., 2014), several QLI classes with 0, 1, or 2 quantifier alternations are analyzed and their complexity is established; polynomial-time procedures can be devised for all sub-classes in **P**. Unbounded QLPs are studied in (Ruggieri et al., 2014), where it is shown that the corresponding decision problem is in **P**. The implication of this result is that the presence of universal quantifiers does not alter the complexity of the linear programming problem. In (Wojciechowski et al., 2015), the same result was also obtained for all partially bounded and unbounded variants of QLP and QLI. Furthermore, it was shown that for each class of the **PH**, there exists a QLI that is complete for that class, a result analogous to that of (Stockmeyer, 1977) but with continuous variables.

Results in QIP are rather scarce, while QII has not been examined before. The decision problem for QIP is **PSPACE-hard**. The hardness follows trivially from the fact that Quantified Satisfiability (QSAT) is **PSPACE-complete**. Various special cases of QIPs have also been analyzed in order to identify subclasses that are tractable. Such subclasses include QIPs in which all constraints exist between at most two variables (referred to as Planar QIPs) (Liang et al., 2013) or QIPs in which all variables are universally quantified (referred to as Box QIPs) (Subramani, 2005).

3 QIP Over Polytopes

We use the standard notation of linear algebra (Schrijver, 1987) to formally present the basic notions of this paper. $\mathbb{R}$ is the set of real numbers. Let capital bold letters $(\mathbf{A}, \mathbf{B}, \ldots)$ denote matrices and small bold letters $(\mathbf{x}, \mathbf{y}, \mathbf{b}, \ldots)$ denote column vectors. Moreover, let $\mathbf{0}$ be the column vector with all elements equal to 0. Furthermore, let $\mathbf{A} \cdot \mathbf{x}$ denote the product of $\mathbf{A}$ and $\mathbf{x}$ and let $\mathbf{x} \cdot \mathbf{y}$ denote the inner product of $\mathbf{x}$ and $\mathbf{y}$. Finally, we assume that the dimensions of vectors and matrices in products are of compatible size.

In traditional IP, all variables are (implicitly) existentially quantified. A Quantified Integer Program (Subramani, 2005) is a conjunctive system of linear constraints with all variables restricted to integer values, in which each variable is either existentially or universally quantified according to a given quantifier string and each universally quantified variable ranges over an interval of integers:

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \in \{\mathbf{l}_1 - \mathbf{u}_1\} \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \in \{\mathbf{l}_n - \mathbf{u}_n\} \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}. \tag{1}$$

In QIP (1), $\mathbf{x}_1 \ldots \mathbf{x}_n$ is a partition of $\mathbf{x}$ with, possibly, $\mathbf{x}_1$ empty; $\mathbf{y}_1 \ldots \mathbf{y}_n$ is a partition of $\mathbf{y}$ with, possibly, $\mathbf{y}_n$ empty; and $\mathbf{l}_i$, $\mathbf{u}_i$ are lower and upper bounds for $\mathbf{y}_i$, $i = 1, \ldots, n$. Note that in a QIP each universally quantified variable is bounded from above and below. We say that a QIP *holds* if it is true as a first-order formula over the domain of the integers. The *decision problem for a QII* consists of checking whether it holds or not.

As stated before, it is trivial to reduce QSAT to QIP. Hence, QIP is **PSPACE-hard**. We now study the case where the linear system of the QIP is a polytope. This means that for each existential variable we can always derive both an upper and lower bound.

Theorem 3.1 *QIP over polytopes is* **PSPACE-complete**.

Proof: We only need to show that QIP over polytopes is in **PSPACE**. Consider the following form of a QIP:

$$\exists x_1 \ \forall y_1 \in \{l_1 - u_1\} \ \ldots \ \exists x_n \ \forall y_n \in \{l_n - u_n\} \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}$$

Note that QIP (1) can be easily converted into this form through the addition of dummy variables.

In this QIP, $\mathbf{y}$ can take $\prod_{i=1}^{n}(u_i - l_i + 1)$ possible values. For each value $\mathbf{y^j}$, we can consider the constraint matrix $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j}$.

Let $\mathbf{A^j} \cdot \mathbf{x} \leq \mathbf{b^j}$ be the constraint matrix representing the integer hull of $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j}$. For each $\mathbf{y^j}$, we can find

$$l_j = \min x_1 \quad \mathbf{A^j} \cdot \mathbf{x} \leq \mathbf{b^j}.$$

We know that a finite l_j exists and that it takes its value at an extreme point of $\mathbf{A^j} \cdot \mathbf{x} \leq \mathbf{b^j}$. Thus, $l_j \in \mathbb{Z}$. Since $\mathbf{A^j} \cdot \mathbf{x} \leq \mathbf{b^j}$ is the integer hull of $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j}$ we also have that

$$\begin{aligned} l_j = \min x_1 & \\ \mathbf{A} \cdot \mathbf{x} &\leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j} \\ \mathbf{x} &\in \mathbb{Z}^n. \end{aligned}$$

Thus, $\left|l_j - l'_j\right| \leq n \cdot \Xi(\mathbf{A})$ where

$$\begin{aligned} l'_j = \min x_1 & \\ \mathbf{A} \cdot \mathbf{x} &\leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j} \end{aligned}$$

and $\Xi(\mathbf{A})$ is the maximum sub-determinant of $\mathbf{A}$ (Cook et al., 1986).

We have that $l'_j \leq l_j$. Thus, $l_j \leq l'_j + \Xi(\mathbf{A})$.

Since l'_j corresponds to the value of x_1 at an extreme point of $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} - \mathbf{M} \cdot \mathbf{y^j}$ we know that $size(l'_j) \leq 4 \cdot (size(\mathbf{A}) + size(\mathbf{b} - \mathbf{M} \cdot \mathbf{y^j}))$. Thus, l'_j and l_j are both polynomially sized in terms of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{M}$ (Cook et al., 1986).

For each $\mathbf{y^j}$ we can also find

$$u_j = \max x_1 \quad \mathbf{A^j} \cdot \mathbf{x} \leq \mathbf{b^j}.$$

Using similar arguments, we can show that u_j always exists and is polynomially sized in terms of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{M}$.

Let $l = \max_{j=1,\ldots,2^n} l_j$ and $u = \min_{j=1,\ldots,2^n} u_j$. We have that both l and u are polynomially sized in terms of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{M}$. Hence, we have the following cases to consider

1. If $l > u$, then the QIP is infeasible.
2. If $l \leq u$, then we only need to consider $l \leq x_1 \leq u$. All such values are polynomially sized.

Thus, we can always restrict x_1 to be polynomially sized.

Since $l_1 \leq y_1 \leq u_1$, we have that all possible values of y_1 are also polynomially sized. Thus, after both x_1 and y_1 are chosen, we get a new QIP with x_2 as the first variable. Using the same arguments we can show that all possible values for x_2 and y_2 are also polynomially sized. Using induction, we can repeat this process for each x_i and y_i. Thus, QIP over polytopes is in **PSPACE**. □

4 Unbounded variants

We now show that the unbounded variants of both problems capture **PA**. This will allow us to show that QII is also **PA-complete**.

4.1 Complexity of UQIP

An *Unbounded Quantified Integer Program* (UQIP) is a QIP that has no bounds on universal variables:

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \ \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}, \tag{2}$$

Theorem 4.1 *UQIP is* **PA-complete**.

Proof: We show that by allowing for unbounded quantifiers UQIP captures **PA**. Recall that **PA** does not allow for multiplication. However, all multiplications in UQIPs are done by known integer constants. Thus, the multiplication in UQIP is simply shorthand for repeated addition. For example, $5 \cdot x_i$ can be considered shorthand for $(x_i + x_i + x_i + x_i + x_i)$.

Let us first show that every atomic formula (no quantifiers or boolean operations) is simply an equality constraint. Let $P(\mathbf{x}) \leq a$ be an atomic formula in **PA**. Note that **PA** does not explicitly allow inequality. However $P(\mathbf{x}) \leq a$ can be expressed in the language of **PA** as:

$$\exists w \ \ P(\mathbf{x}) + w = a.$$

This correctly expresses the desired inequality since all variables in **PA** are non-negative.

Hence, every atomic formula can be represented as an equality constraint in the UQIP framework. Thus, to prove our claim, we only need to show that quantification and boolean operations can be expressed by the UQIP framework.

Let us first examine quantification. Consider the expression $\forall \mathbf{y} \ Q(\mathbf{y}) \leq b$. This can be rewritten as $\forall \mathbf{y} \ \exists w \ Q(\mathbf{y}) +$

$w = b$. Note that w is added at the end of the quantifier string. This is because we do not want $\mathbf{y}$ to depend on the choice for w. The original constraint holds as long as there always exists an appropriate value of w. It is not a requirement that w has the same value in all cases. In fact, the value of w can be different in every case and the original constraint will still hold. Hence, UQIPs already allow for universal quantifiers. The same holds for existentially quantified expressions in **PA**. It is easy to see that $\exists \mathbf{x}\ P(\mathbf{x}) \leq a$ can be transformed to $\exists \mathbf{x}\ \exists w\ P(\mathbf{x}) + w = a$. Thus, any quantified expression in **PA** is already valid in the language of UQIPs.

Let us now examine boolean operations in **PA**.

1. $(P(\mathbf{x}) \leq a) \wedge (Q(\mathbf{y}) \leq b)$: UQIPs already allow for the conjunction of constraints. Thus, this is already valid in the language of UQIPs.
2. $\neg(P(\mathbf{x}) \leq a)$: Recall that $P(\mathbf{x})$ is guaranteed to be integral. Thus, this is equivalent to:
$$a + 1 \leq P(\mathbf{x})$$
which is valid in the language of UQIPs.
3. $(P(\mathbf{x}) \leq a) \vee (Q(\mathbf{y}) \leq b)$: UQIPs do not explicitly allow disjunction. However we can handle this by adding a new integer variable, w, and an associated large coefficient, M. We can then represent this disjunction as:
$$\exists w\ \ (0 \leq w \leq 1) \wedge (P(\mathbf{x}) - M \cdot w \leq a) \wedge \\ (Q(\mathbf{y}) - M \cdot (1 - w) \leq b)$$
In this expression, w is used to choose which statement in the disjunction is forced to hold and which is relaxed. Thus, this is valid in the language of UQIPs. Again, in the presence of quantification, w is added at the end of the quantifier string. That is,
$$\exists \mathbf{x}\ \forall \mathbf{y}\ \ (P(\mathbf{x}) \leq a) \vee (Q(\mathbf{y}) \leq b)$$
becomes
$$\exists \mathbf{x}\ \forall \mathbf{y}\ \exists w\ \ (0 \leq w \leq 1) \wedge (P(\mathbf{x}) - M \cdot w \leq a) \wedge \\ (Q(\mathbf{y}) - M \cdot (1 - w) \leq b).$$
Note that for the disjunction to be satisfied only one of the original constraints needs to be satisfied. However, which constraint is satisfied depends on the choices of $\mathbf{x}$ and $\mathbf{y}$. Thus, it is necessary for w to be chosen after $\mathbf{y}$. If $\mathbf{y}$ were chosen to violate both of the original constraints, then both choices for w would result in a violated constraint, thus correctly simulating the disjunction.

Hence, through the use of these transformations and De Morgan's Laws we can transform any statement in **PA** into a UQIP. Therefore, UQIP is **PA-hard**. To finalize the proof, notice that UQIP is a special case of **PA**. Thus UQIP is **PA-complete**. □

Note that UQIP is **PA-complete** while QIP is **PSPACE-complete**. This difference lies in the fact that in QIPs all universally quantified variables are guaranteed to be bounded.

4.2 Complexity of UQII and QII

A Quantified Integer Implication extends the quantification of integer variables to implications of two linear constraint systems:

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \\ [\mathbf{C} \cdot \mathbf{x} + \mathbf{N} \cdot \mathbf{y} \leq \mathbf{b} \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{d}] \qquad (3)$$

where $\mathbf{x}_1 \ldots \mathbf{x}_n$ is a partition of $\mathbf{x}$ with, possibly, $\mathbf{x}_1$ empty and $\mathbf{y}_1 \ldots \mathbf{y}_n$ is a partition of $\mathbf{y}$ with, possibly, $\mathbf{y}_n$ empty. We say that a QII *holds* if it is true as a first-order formula over the domain of the integers. The *decision problem for a QII* consists of checking whether it holds or not.

An *Unbounded Quantified Integer Implication* (UQII) is a QII that includes only existential variables in its LHS (hence not 'bounding' the universal variables):

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \\ [\mathbf{C} \cdot \mathbf{x} \leq \mathbf{d} \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}] \qquad (4)$$

The following theorem establishes the computational complexity of UQII.

Theorem 4.2 *UQII is* **PA-complete**.

Proof: We reduce a UQIP instance of the form (2), which is **PA-complete** (Theorem 4.1), to a UQII instance of the form (4). We construct the UQII as follows:

1. For every vector $\mathbf{x}_i$ of UQIP (2), we add a vector $\mathbf{x}_i$ to the corresponding UQII.
2. For every vector $\mathbf{y}_i$ of UQIP (2), we add a vector $\mathbf{y}_i$ to the corresponding UQII.
3. All the constraints in $\mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}$ are added to the RHS of the implication.
4. To represent the unbounded universal variables, we add the constraint $0 \leq 1$ to the LHS of the implication.

Furthermore, we use the same quantifier string as UQIP (2). Hence, the constructed system will be as follows:

$$\exists \mathbf{x}_1\ \forall \mathbf{y}_1\ \ldots\ \exists \mathbf{x}_n\ \forall \mathbf{y}_n\ \ [0 \leq 1 \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}]$$

This system is clearly of the form (4). Since the LHS of the implication is always true, the implication is satisfied if and only if $\mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}$. Thus, the constructed UQII is feasible if and only if

$$\exists \mathbf{x}_1\ \forall \mathbf{y}_1\ \ldots\ \exists \mathbf{x}_n\ \forall \mathbf{y}_n\ \ \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}.$$

is feasible. Thus UQII is **PA-hard**. Moreover, just like UQIP, UQII is a special case of **PA**. Thus UQII is **PA-complete**. □

We can now easily extend this result to establish the computational complexity of QII.

Corollary 4.1 *QII is* **PA-complete**.

Proof: By definition, a UQII is a QII. Thus, QII is **PA-hard**. Moreover, just like UQIP, QII is a special case of **PA**. Thus QII is **PA-complete**. □

5 Partially bounded variants

In this section, we focus on problems where the universally quantified variables are only restricted from one side. Without loss of generality, we can assume that these variables are only bounded from below.

5.1 Complexity of PQIP

A *Partially bounded Quantified Integer Program* (PQIP) is a QIP in which each universally quantified variable is only bounded on one side. Without loss of generality, we can assume this single bound forces each such variable to be non-negative.

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \in \{\mathbf{0} - \infty\} \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \in \{\mathbf{0} - \infty\} \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b} \tag{5}$$

This assumption holds because we can always transform the PQIP into the desired form as demonstrated in the example that follows.

Example (1): Consider PQIP (6):

$$\exists x_1 \ \forall y_1 \in (-\infty, -5] \ \ x_1 + y_1 \leq 2. \tag{6}$$

We can change the upper bound on y_1 into a non-negativity lower bound by replacing all instances of y_1 in PQIP (6) with $-(y_1 + 5)$. This results in PQIP (7):

$$\exists x_1 \ \forall y_1 \in [0, +\infty) \ \ x_1 - (y_1 + 5) \leq 2. \tag{7}$$

Theorem 5.1 *PQIP is* **PA-complete**.

Proof: We reduce a UQIP instance of the form (2) to a PQIP instance of the form (5). We construct the PQIP as follows:

1. For every vector $\mathbf{x}_i$ of UQIP (2), we add a vector $\mathbf{x}_i$ to the corresponding PQIP.
2. For every vector $\mathbf{y}_i$ of UQIP (2), we add vectors $\mathbf{y}_i' \in \{\mathbf{0} - \infty\}$ and $\mathbf{y}_i'' \in \{\mathbf{0} - \infty\}$ to the corresponding PQIP.
3. Let $\mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot (\mathbf{y}' - \mathbf{y}'') \leq \mathbf{b}$ denote the constraints of the PQIP.

Furthermore, we construct the quantifier string similarly to that of the UQIP, replacing each $\mathbf{y}_i$ of UQIP (2) with its corresponding vectors $\mathbf{y}_i'$ and $\mathbf{y}_i''$. Hence, the constructed system will be as follows:

$$\exists \mathbf{x}_1 \ \forall \mathbf{y}_1' \in \{\mathbf{0} - \infty\} \ \forall \mathbf{y}_1'' \in \{\mathbf{0} - \infty\} \ \ldots \ \exists \mathbf{x}_n \ \forall \mathbf{y}_n' \in \{\mathbf{0} - \infty\} \ \forall \mathbf{y}_n'' \in \{\mathbf{0} - \infty\} \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot (\mathbf{y}' - \mathbf{y}'') \leq \mathbf{b}$$

This system is clearly of the form (5). Also, this system is feasible if and only if

$$\exists \mathbf{x}_1 \ \forall \mathbf{y}_1 \ \ldots \ \exists \mathbf{x}_n \ \forall \mathbf{y}_n \ \ \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}.$$

is feasible. Thus, PQIPs are **PA-hard**. Moreover, just like UQIP, PQIP is a special case of **PA**. Hence, PQIPs are **PA-complete**. □

5.2 Complexity of PQII

A *Partially bounded Quantified Integer Implication* (PQII) is a QII whose universally quantified variables are bounded on one side. As with PQIPs, without loss of generality, we can assume this single bound forces each such variable to be non-negative. Note that these constraints are not included in the quantifier string (as in the QIP or UQIP case) but in the *Left-Hand Side* (LHS) of the corresponding constraints.

$$\exists \mathbf{x}_1 \forall \mathbf{y}_1 \ldots \exists \mathbf{x}_n \forall \mathbf{y}_n \quad [\mathbf{C} \cdot \mathbf{x} \leq \mathbf{d}, \mathbf{y} \geq \mathbf{0} \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}] \tag{8}$$

Theorem 5.2 *PQII is* **PA-complete**.

Proof: We reduce a PQIP instance of the form (5), which is **PA-complete** (Theorem 5.1), to a PQII instance of the form (8). We construct the PQII as follows:

1. For every vector $\mathbf{x}_i$ of PQIP (5), we add a vector $\mathbf{x}_i$ to the corresponding PQII.
2. For every vector $\mathbf{y}_i$ of PQIP (5), we add a vector $\mathbf{y}_i$ to the corresponding PQII.
3. All the constraints in $\mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}$ are added to the RHS of the implication.
4. To represent the bounds on the universally quantified variables, we add the constraints $\mathbf{y} \geq \mathbf{0}$ to the LHS of the implication.

Furthermore, we utilize the same quantifier string as the PQIP to obtain the following system:

$$\exists \mathbf{x}_1 \ \forall \mathbf{y}_1 \ \ldots \ \exists \mathbf{x}_n \ \forall \mathbf{y}_n \ \ [\mathbf{y} \geq \mathbf{0} \rightarrow \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}]$$

This system is clearly of the form (8). Let y_i^j be the jth variable of $\mathbf{y}_i$. If for some universally quantified variable y_i^j we have $y_i^j < 0$, then this system is automatically satisfied. This means that, without loss of generality, we can restrict any universally quantified variable y_i^j to the set $\{0 - \infty\}$. Thus, the constructed PQII is feasible if and only if

$$\exists \mathbf{x}_1 \ \forall \mathbf{y}_1 \in \{\mathbf{0} - \infty\} \ \ldots \ \exists \mathbf{x}_n \ \forall \mathbf{y}_n \in \{\mathbf{0} - \infty\} \quad \mathbf{A} \cdot \mathbf{x} + \mathbf{M} \cdot \mathbf{y} \leq \mathbf{b}.$$

is feasible. Therefore, PQIIs are **PA-hard**. Moreover, just like UQIP, PQII is a special case of **PA**. Hence, PQIIs are **PA-complete**. □

6 QIP and the PH

For the sake of completeness, we present the following results on the relation of QIPs and QIIs with the **PH**. In this section, we examine the relation of each level of the **PH** with QIPs with limited quantifier alternations.

Theorem 6.1
$\forall \mathbf{y}_1 \in \{\mathbf{l}_1 - \mathbf{u}_1\} \ \exists \mathbf{x}_1 \ \ldots \ \forall \mathbf{y}_k \in \{\mathbf{l}_k - \mathbf{u}_k\} \ \exists \mathbf{x}_k$
$\mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{y} \leq \mathbf{c}$ *is* $\mathbf{\Pi_P^{2 \cdot k}}$**-hard**.

Proof: We start with the following restricted form of Q3SAT.

$$\forall \mathbf{y}_1 \ \exists \mathbf{x}_1 \ \ldots \ \forall \mathbf{y}_k \ \exists \mathbf{x}_k \ \ \phi(\mathbf{x}, \mathbf{y})$$

Note that this problem is $\mathbf{\Pi_P^{2 \cdot k}}$**-complete**. We construct an instance of QIP as follows:

1. For each vector $\mathbf{x}_i$, we add the vector $\mathbf{x}_i$ and the constraints $\mathbf{0} \leq \mathbf{x}_i \leq \mathbf{1}$ to the system.
2. For each vector $\mathbf{y}_i$, we add the vector $\mathbf{y}_i$ to the system with $\{\mathbf{0} - \mathbf{1}\}$ bound in the quantifier string.
3. For each 3CNF clause, c_k, in ϕ, we add a constraint to the QIP. This constraint depends on the form of the clause. For example, the clause $(y_1, \neg y_2, x_3)$ is represented by the constraint

$$y_1 + (1 - y_2) + x_3 \geq 1.$$

The quantifier string of the constructed QIP instance is of the form:

$$\forall \mathbf{y}_1 \in \{\mathbf{0}-\mathbf{1}\}\ \exists \mathbf{x}_1\ \ldots\ \forall \mathbf{y}_k \in \{\mathbf{0}-\mathbf{1}\}\ \exists \mathbf{x}_k$$

If a variable in the Q3SAT instance is given a value of **true**, then the value 1 is assigned to the corresponding integer variable in the QIP. On the other hand, if a variable in the Q3SAT instance is given a value of **false**, then the value 0 is assigned to the corresponding integer variable in the QIP.

First, consider a case where the Q3SAT instance holds. This means that every clause ϕ_k in ϕ is **true**, which in turn means that at least one of the literals in ϕ_k is **true**. Therefore, one of the variables in the corresponding constraint will take the value 1. Hence, the corresponding constraint will be trivially satisfied. Since all clauses are satisfied, all constraints of the corresponding QIP will also be satisfied, thus the QIP will also hold.

Now consider a case where the Q3SAT instance does not hold. This means that for every assignment, at least one clause ϕ_k in ϕ is **false**, which in turn means that all of its literals are **false**. Therefore, the corresponding constraint will not be satisfied, since all its variables will take the value 0. Since, for every variable assignment, at least one constraint of the corresponding QIP will not be satisfied, the QIP will also not hold. □

In a similar manner, the following result can be obtained for QIPs starting with an existential quantifier.

Theorem 6.2
$\exists \mathbf{x}_1\ \forall \mathbf{y}_1 \in \{\mathbf{l}_1 - \mathbf{u}_1\}\ \ldots\ \forall \mathbf{y}_k \in \{\mathbf{l}_k - \mathbf{u}_k\}\ \exists \mathbf{x}_{k+1}$
$\mathbf{A}\cdot\mathbf{x} + \mathbf{B}\cdot\mathbf{y} \le \mathbf{c}$ *is* $\mathbf{\Sigma_P^{2\cdot k+1}}$**-hard**.

It is easy to obtain results similar to Theorems 6.1 and 6.2 for the unbounded and partially bounded variants of QIP as well.

7 QII and the PH

Based on the results of Section 6, we can now use the relationship between QIP and QII to establish similar results for QII and its variants and their relation to each level of the **PH**.

Theorem 7.1 $\forall \mathbf{y}_1\ \exists \mathbf{x}_1\ \ldots\ \forall \mathbf{y}_k\ \exists \mathbf{x}_k$
$[\mathbf{C}\cdot\mathbf{x} + \mathbf{N}\cdot\mathbf{y} \le \mathbf{d} \rightarrow \mathbf{A}\cdot\mathbf{x} + \mathbf{M}\cdot\mathbf{y} \le \mathbf{b}]$ *is* $\mathbf{\Pi_P^{2\cdot k}}$**-hard**.

Proof: Consider a QIP instance of the form:

$$\forall \mathbf{y}_1 \in \{\mathbf{l}_1 - \mathbf{u}_1\}\ \exists \mathbf{x}_1\ \ldots\ \forall \mathbf{y}_k \in \{\mathbf{l}_k - \mathbf{u}_k\}\ \exists \mathbf{x}_k \quad \mathbf{A}\cdot\mathbf{x} + \mathbf{B}\cdot\mathbf{y} \le \mathbf{c}$$

From Theorem 6.1 we have that this problem is $\mathbf{\Pi_P^{2\cdot k}}$**-hard**. We construct the corresponding QII as follows:

1. For every vector $\mathbf{x}_i$ of the QIP, we add a vector $\mathbf{x}_i$ to the corresponding QII.
2. For every vector $\mathbf{y}_i$ of the QIP, we add a vector $\mathbf{y}_i$ to the corresponding QII.
3. All the constraints in $\mathbf{A}\cdot\mathbf{x} + \mathbf{M}\cdot\mathbf{y} \le \mathbf{b}$ are added to the RHS of the implication.
4. To represent the bounds on the universally quantified variables, we add the constraints $\mathbf{l} \le \mathbf{y} \le \mathbf{u}$ to the LHS of the implication.

The quantifier string of the constructed QII will have the following form:

$$\forall \mathbf{y}_1\ \exists \mathbf{x}_1\ \ldots\ \forall \mathbf{y}_k\ \exists \mathbf{x}_k$$

Thus, the constructed system will be as follows:

$$\forall \mathbf{y}_1\ \exists \mathbf{x}_1\ \ldots\ \forall \mathbf{y}_k\ \exists \mathbf{x}_k\ \ [\mathbf{l} \le \mathbf{y} \le \mathbf{u} \rightarrow \mathbf{A}\cdot\mathbf{x} + \mathbf{M}\cdot\mathbf{y} \le \mathbf{b}]$$

This system is clearly of the form (3). The result follows. □

The following result on QIIs starting with an existential quantifier is obtained similarly.

Theorem 7.2 $\exists \mathbf{x}_1\ \forall \mathbf{y}_1\ \ldots\ \exists \mathbf{x}_k\ \forall \mathbf{y}_k\ \exists \mathbf{x}_{k+1}$
$[\mathbf{C}\cdot\mathbf{x} + \mathbf{N}\cdot\mathbf{y} \le \mathbf{d} \rightarrow \mathbf{A}\cdot\mathbf{x} + \mathbf{M}\cdot\mathbf{y} \le \mathbf{b}]$ *is* $\mathbf{\Sigma_P^{2\cdot k+1}}$**-hard**.

We can easily obtain analogous results for the unbounded and partially bounded variants of QII as well.

8 Conclusion

In this paper, we introduced several variants of QIP and QII and discussed their computational complexities. We have shown that QIP over polytopes is **PSPACE-complete**. Additionally, we have shown that partially bounded and unbounded variants of QIPs and QIIs are **PA-complete**. This is in contrast to the corresponding variations of QLPs and QLIs, which are in **P**. Intuitively, this occurs because in the 'integer" problems there are no implied restrictions to the values that universal variables may assume. Finally, we have examined the connections between alternations in the quantifier strings of these problems and levels of the polynomial hierarchy.

Avenues for future research include establishing the computational complexity of QIP. Furthermore, showing that each level of the **PH** can be represented by QIPs with limited quantifier alternations is an interesting open issue.

A Presburger Arithmetic

The theory of Presburger Arithmetic has the signature:

$$\{0, 1, +, =\}$$

where

(i) 0 and 1 are constants.

(ii) $+$ is a binary function.

(iii) $=$ is a binary predicate.

Its axiom set is the following:

(a) $(\forall x)\ \neg(x+1) = 0$

(b) $(\forall x)\ (\forall y)\ [(x+1) = (y+1)] \rightarrow (x = y)$

(c) $(\forall x)\ (x+0 = x)$

(d) $(\forall x)\ (\forall y)\ x + (y+1) = (x+y)+1$

(e) $(F[0] \wedge (\forall x)\ (F[x] \rightarrow F[x+1])) \rightarrow (\forall y)\ F[y]$

References

L. Berman. The complexitiy of logical theories. *Theoretical Computer Science*, 11:71–77, 1980.

F. Benhamou and F. Goulard. Universally quantified interval constraints. In *Principles and Practice of Constraint Programming (CP 2000)*, pages 67–82. Springer Berlin Heidelberg, 2000.

A.R. Bradley and Z. Manna. *The calculus of computation - decision procedures with applications to verification*. Springer, 1^{st} edition, 2007.

C.W. Brown. QEPCAD B: A program for computing with semi-algebraic sets using CADs. *ACM SIGSAM Bullettin*, 37(4):97–108, 2003.

G.E. Collins and H. Hong. Partial cylindrical algebraic decomposition for quantifier elimination. *Journal of Symbolic Computation*, 12(3):299–328, 1991.

A. Colmerauer. Expressiveness of full first-order constraints in the algebra of finite or infinite trees. *Constraints*, 8(3):283–302, 2003.

W. Cook, A.M. Gerards, A. Schrijver, E. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34(3):251–264, 1986.

A. Dolzmann and T. Sturm. REDLOG: Computer algebra meets computer logic. *ACM SIGSAM Bullettin*, 31(2):2–9, 1997.

A. Dolzmann, T. Sturm, and V. Weispfenning. A new approach for automatic theorem proving in real geometry. *Journal of Automated Reasoning*, 21(3):357–380, 1998.

P. Eirinakis, S. Ruggieri, K. Subramani, and P. Wojciechowski. A complexity perspective on entailment of parameterized linear constraints. *Constraints*, 17(4):461–487, 2012.

P. Eirinakis, S. Ruggieri, K. Subramani, and P. Wojciechowski. On quantified linear implication. *Annals of Mathematics and Artificial Intelligence*, 71(4):301–325, 2014.

T. Ferguson. *Lecture Notes in Game Theory.* University of California, Los Angeles, 2013.

M.J. Fischer and M.O. Rabin. Super-exponential complexity of presburger arithmetic. Technical Report Project MAC Technical Memorandum 43, MIT, Cambridge, MA, 1974.

J. Ferrante and C. Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing*, 4(1):69–76, 1975.

R. Gerber, W. Pugh, and M. Saksena. Parametric dispatching of hard real-time tasks. *IEEE Transactions on Computing*, 44(3):471–479, 1995.

S. Iwata, T. Matsui, and S.T. McCormick. A fast bipartite network flow algorithm for selective assembly. *Operations Research Letters*, 22:137–143, 1998.

C. Lassez and J.-L. Lassez. *Quantifier Elimination for Conjunctions of Linear Constraints via a Convex Hull Algorithm*. Academic Press, New York, NY, 1993.

J.-L. Lassez and M. Maher. On fourier's algorithm for linear arithmetic constraints. *Journal of Automated Reasoning*, 9(3):373–379, 1992.

Z. Liang, K. Subramani, and J. Worthington. Polynomial time certifying algorithms for the planar quantified integer programming problem. *Journal of Logic and Computation*, 23(5):1017–1033, 2013.

S.T. McCormick. Fast algorithms for parametric scheduling come from extensions to parametric maximum flow. *Operations Research*, 47:744–756, 2000.

C.H. Papadimitriou. *Computational Complexity*. Addison-Wesley, New York, 1994.

S. Ratschan. Efficient solving of quantified inequality constraints over the real numbers. *ACM Transactions on Computational Logic*, 7(4):723–748, 2006.

S. Ruggieri, P. Eirinakis, K. Subramani, and P. Wojciechowski. On the complexity of quantified linear systems. *Theoretical Computer Science*, 518:128–134, 2014.

A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley and Sons, New York, 1987.

P. Serafini. Scheduling jobs on several machines with the job splitting property. *Operations Research*, 44:617–628, 1996.

J.A. Stankovic, M. Spuri, K. Ramamritham, and G.C. Buttazzo. *Deadline Scheduling for Real-Time Systems*. Kluwer Academic, Dordrecht, 1998.

L.J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.

K. Subramani. An analysis of partially clairvoyant scheduling. *Journal of Mathematical Modelling and Algorithms*, 2(2):97–119, 2003.

K. Subramani. Tractable fragments of presburger arithmetic. *Theory of Computing Systems*, 38(5):647–668, 2005.

K. Subramani. On a decision procedure for quantified linear programs. *Annals of Mathematics and Artificial Intelligence*, 51(1):55–77, 2007.

A. Tarski. A decision method for elementary algebra and geometry. Technical Report R-109, Rand Corporation, 1948.

V. Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 4(1-2):3–27, 1988.

P. Wojciechowski, P. Eirinakis, and K. Subramani. Analyzing restricted fragments of the theory of linear arithmetic. *Annals of Mathematics and Artificial Intelligence* (accepted). DOI: 10.1007/s10472-015-9489-z