Optimization, Games, and Quantified Constraint Satisfaction

Hubie Chen and Martin Pál *

Department of Computer Science
Cornell University
Ithaca, NY 14853, USA
{hubes,mpal}@cs.cornell.edu

Abstract. Optimization problems considered in the literature generally assume a passive environment that does not react to the actions of an agent. In this paper, we introduce and study a class of optimization problems in which the environment plays an active, adversarial role and responds dynamically to the actions of an agent; this class of problems is based on the framework of quantified constraint satisfaction. We formalize a new notion of approximation algorithm for these optimization problems, and consider certain restricted versions of the general problem obtained by restricting the types of constraints that may appear. Our main result is a dichotomy theorem classifying exactly those restricted versions having a constant factor approximation algorithm.

1 Introduction

Optimization is what we do all of the time. We strive for better products, faster cars, more efficient networks, and cheaper solutions. Research in both continuous and discrete optimization has seen tremendous growth over the past two decades. It has long been known that many problems of interest are intractable to solve to optimality; this fact has been the driving force behind the study of approximation algorithms.

Most optimization problems considered by researchers assume a passive environment that does not react to the actions of an agent. The state of the environment is assumed to be fully known (deterministic optimization), partially random (stochastic optimization), or unknown in advance but independent of an agent's actions (on-line algorithms). In this paper, we introduce and study a class of optimization problems in which the environment plays an active role. In contrast to previously considered optimization problems, the problems we study are capable of modelling scenarios where the environment actively responds to the moves of an agent in an adversarial manner.

The framework of problems that we study is based on the *quantified constraint satisfaction problem* (QCSP). An instance of the QCSP consists of a quantified formula, which is a set of constraints along with an ordered list of

* Research supported by ONR grant N00014-98-1-0589.

decision variables, each having an existential or universal quantifier associated with it. A quantified formula can be viewed as a game between two players, a *universal player*, who strives to falsify a constraint, and an *existential player*, who attempts to satisfy all constraints; these players set their variables in the specified order, and the existential player is said to have a *winning strategy* if she can always succeed in satisfying all of the constraints, despite the moves of the universal player. In the vanilla QCSP, the problem is to decide whether or not the existential player has a winning strategy. In our framework, which generalizes the QCSP, the existential player strives to satisfy all constraints and, subject to this, minimize an objective function. No assumptions are made about the environment (the universal player), which may set variables in an adversarial manner, trying to invalidate a constraint or worsen the objective function value.

We formalize and prove results concerning a new notion of approximation algorithm for these game-based minimization problems. How do we formalize this notion? When universal quantification is removed from the described games, a strategy is simply a static assignment to the variables, and is winning if it satisfies all of the constraints. The established notion of approximation algorithm for this special case is a polynomial-time algorithm that always computes an assignment satisfying all constraints, when such an assignment exists; such an algorithm is said to achieve approximation ratio α if on every satisfiable problem instance, the computed assignment has cost no more than α times the minimum cost assignment. When we include universal quantification, an approximation algorithm still ought to efficiently generate a winning strategy. However, we impose computational bounds not just on the generation of the strategy, but also on the generated strategy itself, in order to ensure that it is effectively usable. (A strategy specified succinctly as the description of a Turing machine having an exponential running time is an example of a strategy that should not be considered acceptable.) We therefore mandate that an approximation algorithm output a winning strategy (when one exists) specified as a boolean circuit, in polynomial time; such a circuit, of course, is necessarily of polynomial size, and hence can itself be evaluated in polynomial time. We take the (arguably pessimistic!) approach of defining the cost of a strategy to be the maximum cost assignment it produces, over all adversaries; that is, a strategy is graded according to its worst adversary.[1] An approximation algorithm is considered to achieve approximation ratio α if on every problem instance (with a winning strategy), the computed strategy has cost no more than α times the minimum cost winning strategy. It is straightforward to verify that this notion of approximation algorithm generalizes the discussed, established notion for the case where there is no universal quantification.

The particular class of problems we study in this paper is as follows. We investigate quantified formulas where all decision variables are over the boolean domain $\{0, 1\}$ and have weights associated with them, and the objective function to be minimized is the total weight of the variables set to 1. We search for

[1] We leave the investigation of other strategy measures as an interesting issue for future work.

constant factor approximable classes of these formulas by studying restricted classes of quantified formulas where the types of constraints that may appear must come from a fixed *constraint language*. This form of restriction was first studied by Schaefer [9] and has since been investigated heavily, in several different contexts (see [5] for examples); it is widely acknowledged as a useful and rich means of obtaining restricted classes of the constraint satisfaction problem and its variants. Khanna et al. [8] have studied the non-quantified versions of the class of problems we consider here, and obtained a theorem classifying, for all constraint languages S, the approximability of such problems having constraints built using S. Their motivation was to systematically classify and unify the many existing results on minimization problems.

Our main result is a full dichotomy theorem characterizing the constraint languages for which a constant factor approximation exists for our QCSP minimization problems. Interestingly, every non-trivial constraint language for which there is a constant factor approximation in the non-quantified case is also constant factor approximable in the quantified case. Since an α-approximation algorithm for the quantified case is also an α-approximation algorithm for the non-quantified case, our results strictly generalize the constant factor approximation results given in [8].

2 Preliminaries

Let $[n]$ denote the set containing the first n positive integers, $\{1, \dots, n\}$.

A subset of $\{0,1\}^k$ (for some $k \geq 1$) is a *logical relation* of arity k. A *constraint* is an expression of the form $R(w_1, \dots, w_k)$ where R is a logical relation of arity k and the w_i are variables; it is satisfied by a mapping $f : V \to \{0,1\}$ defined on all of its variables if $(f(w_1), \dots, f(w_k)) \in R$.

A *quantified formula* is an expression of the form $\phi = Q_1 v_1 \dots Q_n v_n \mathcal{C}$ where for all $i \in [n]$, Q_i is quantifier from the set $\{\exists, \forall\}$ and v_i is a variable; and, $\mathcal{C}$ is a set of constraints over the variable set $V = \{v_1, \dots, v_n\}$. A weighted quantified formula is a pair $\Phi = (\phi, w)$, where $w : V \to \mathbb{N}$ is a mapping assigning every variable of ϕ a nonnegative weight.

Let $|\phi|_\exists$ ($|\phi|_\forall$) denote the number of existentially (universally) quantified variables in a quantified formula ϕ. For all $i \in [|\phi|_\exists]$, let Y_i^ϕ denote the set of universally quantified variables coming before the ith existentially quantified variable (in the quantification order of ϕ), and for all $j \in [|\phi|_\forall]$, define X_j^ϕ dually.

A quantified formula can be viewed as a two-player game where an existential player and universal player set their variables in the specified order; the existential player wins if all constraints are satisfied, and the universal player wins if some constraint is falsified. Formally, define an existential strategy (for ϕ) to be a sequence of mappings $\Sigma = \{\sigma_i : \{0,1\}^{Y_i^\phi} \to \{0,1\}\}_{i \in [|\phi|_\exists]}$, and similarly define a universal strategy (for ϕ) to be a sequence of mappings $\Pi = \{\pi_i : \{0,1\}^{X_i^\phi} \to \{0,1\}\}_{i \in [|\phi|_\forall]}$. Define a *game* to be a triple (Φ, Σ, Π) where $\Phi = (\phi, w)$ is a weighted quantified formula and Σ (Π) is an existential strategy (universal strategy) for ϕ. The *outcome* of a game $G = (\Phi, \Sigma, \Pi)$ is is the assignment $A_G : V \to \{0,1\}$ to the variable set V of the formula defined by setting the variables in the order specified by ϕ, according to the strategies Σ and Π. A game $G = (\Phi, \Sigma, \Pi)$ is a *win* if the outcome $A_G : V \to \{0,1\}$ satisfies every constraint $R(w_1, \ldots, w_k)$ in ϕ. An existential strategy Σ is *winning* for a formula Φ if the game (Φ, Σ, Π) is a win for all universal strategies Π. A quantified formula is *satisfiable* if there is a winning strategy for it.

The *cost* of a game G is the total weight of all variables set to true, that is, $\mathsf{cost}(G) = \sum_{A_G(v)=1} w(v)$. In the problem that we will consider, the goal of the existential player is to win the game, and subject to this, minimize the cost of the game. The *cost* of a winning existential strategy Σ on weighted formula Φ is defined to be $\mathsf{cost}(\Phi, \Sigma) = \max_{\Pi} \mathsf{cost}(\Phi, \Sigma, \Pi)$, that is, the cost of the game where Σ plays a "worst possible" adversary. Note that there is always an *ignorant strategy* Π maximizing the right-hand side expression, where an ignorant strategy is a strategy that contains only constant functions and does not react to its opponent. Finally, the cost of a weighted formula Φ is defined to be $\mathsf{cost}(\Phi) = \min_{\Sigma} \mathsf{cost}(\Phi, \Sigma)$, where the minimum is taken over all winning existential strategies Σ. That is, the cost of the formula is the cost of the best existential strategy for the formula.

3 Dichotomy theorem

As discussed in the introduction, the problem we study is that of computing, given a weighted quantified formula Φ, an existential strategy Σ that minimizes $\mathsf{cost}(\Phi, \Sigma)$. We consider restricted versions of this problem where the constraints present in the input quantified formulas must come from a specified *constraint language*, where a constraint language is defined to be a set of logical relations. Formally, when S is a constraint language, we define an instance of the $\mathsf{MIN\text{-}QCSP}(S)$ problem to be a weighted quantified formula Φ where all of the constraints in Φ have relation from S. We say that $\mathsf{MIN\text{-}QCSP}(S)$ is α-*approximable* if there exists a polynomial time algorithm that on all satisfiable instances Φ outputs a winning existential strategy Σ (in the form of a sequence of circuits) such that $\mathsf{cost}(\Phi, \Sigma) \leq \alpha \cdot \mathsf{cost}(\Phi)$.

Our main result is a dichotomy theorem on constraint languages, describing exactly those constraint languages S for which $\mathsf{MIN\text{-}QCSP}(S)$ is constant factor approximable. In order to formulate our dichotomy theorem, we will need to define various classes of constraint languages. These definitions are fairly standard [5]. A constraint is *weakly negative* if it is expressible as a CNF-formula having at most one unnegated variable in each clause; *width-2 affine* if it is expressible as a system of linear equations over the two-element field with at most two variables per equation; *bijunctive* if it is expressible as a 2-CNF formula; and *k-IHS-B+* if it is expressible as a CNF formula where each clause is of the form $\neg v$, $\neg v \vee w$, or $w_1 \vee \ldots \vee w_k$. We extend these definitions to constraint languages by stating, for instance, that a constraint language S is weakly negative if any constraint with relation from S is weakly negative. Also, we will state that a quantified formula is weakly negative (bijunctive, etc.) if all of its constraints are weakly negative (bijunctive, etc.).

Theorem 1. *Let S be a constraint language.*

1. *If S is weakly negative or width-2 affine, then* MIN-QCSP(S) *is* 1*-approximable.*
2. *Else if S is bijunctive, then* MIN-QCSP(S) *is* 2*-approximable.*
3. *Else if S is k-IHS-B+, then* MIN-QCSP(S) *is k-approximable.*
4. *Else* MIN-QCSP(S) *is not constant factor approximable, unless* NP *has deterministic algorithms running in time $n^{\log^{O(1)} n}$.*

4 Approximation algorithms and strategies

In this section, we give the positive approximability results of our dichotomy theorem. We first give the results for the cases where we have a constant factor approximation but no 1-approximation (that is, the bijunctive and k-IHS-B+ cases), and then give the weakly negative case. (The width-2 affine case is straightforward, and omitted.) Note that in each of the cases, we do not discuss extensively how to actually implement a polynomial-time algorithm which outputs the desired circuits, and leave it to the reader to convince herself that this is possible.

We will use the following notation throughout this section. When $\phi = Qv\phi'$ is a quantified formula and $b \in \{0, 1\}$, let $\phi[v \leftarrow b]$ denote the formula obtained by substituting all occurrences of the variable v with b in ϕ. Observe that if the formula $\phi = Qv\phi'$ is bijunctive, then $\phi[v \leftarrow b]$ is also bijunctive (and likewise for k-IHS-B+ and weakly negative formulas). Extending this notation, when $\Phi = (\phi, w)$ is a weighted quantified formula, we use $\Phi[v \leftarrow b]$ to denote the weighted quantified formula consisting of $\phi[v \leftarrow b]$ together with the weight function w restricted to $V - \{v\}$. Also, when $\Phi = (\phi, w)$ is a weighted quantified formula, we extend the weight function w to negations of variables by defining $w(\neg v) = 0$, and then to sets of literals S by defining $w(S) = \sum_{l \in S} w(l)$. (Recall that a literal is a variable v or the negation of a variable $\neg v$.)

4.1 The bijunctive case

By definition, any bijunctive constraint is equivalent to a 2-CNF formula. Hence, without loss of generality we assume that all constraints in a bijunctive formula have the form $u \Rightarrow v$ where u and v are literals. We define the implication graph $G(\phi)$ of a bijunctive formula ϕ with variable set V to be the directed graph with $2|V|$ vertices, containing a vertex for each variable $v \in V$ and a vertex for each negated variable $\neg v$. For each constraint $u \Rightarrow v$, it has two edges, $u \rightarrow v$ and $\neg v \rightarrow \neg u$. The closure $T(v)$ of a vertex v in $G(\phi)$ is simply the set of vertices reachable from v via a directed path; we consider the vertex v to be reachable from itself.

To give a strategy for this case, it will be useful to have a subroutine that decides whether or not a quantified formula is satisfiable. This decision can be performed in polynomial time, and hence by polynomial size circuits.

Theorem 2. *[1] The quantified bijunctive formula $\phi = Q_1 v_1 \ldots Q_n \mathcal{C}$ is satisfiable if and only if none of the following three conditions hold for the graph $G(\phi)$:*

1. *An existential literal l is in the same strongly connected component as $\neg l$.*
2. *A universal literal l is in the same strongly connected component as some existential vertex l' such that the variable of l' comes before the variable of l in the quantification order of ϕ.*
3. *There is a path from a universal literal l to another universal literal l' (this includes the possibility $l = \neg l'$).*

It is straightforward to verify that the conditions of Theorem 2 can be checked in polynomial time.

The strategy We give a description of the strategy, which we denote by Σ_0. As the game progresses and variables are being set, Σ_0 transforms the formula Φ by replacing variables with the corresponding values. When it is due to make a move and set a variable v, the strategy Σ_0 first checks if both values $\{0,1\}$ are feasible–that is, if both $\phi[v \leftarrow 1]$ and $\phi[v \leftarrow 0]$ are true. If only one value is feasible, v is set to that value. Otherwise, a decision has to be made.

The strategy strives to set the variable v to the value that minimizes the "implied" costs. Suppose that Σ_0 is about to set a variable v and that the weight of all vertices "implied by" setting v to 1 is zero, that is, $w(T(v)) = 0$. Assuming that $\phi[v \leftarrow 1]$ is true, there is no harm in setting v to 1, as this choice does not commit the algorithm to paying any charges later in the game. Similarly, if $\phi[v \leftarrow 0]$ is true, and $w(T(\neg v)) = 0$, an optimum move is to set v to false.

In case $\phi[v \leftarrow b]$ is true for $b \in \{0,1\}$, we simultaneously decrease the weight of vertices in $T(v)$ and $T(\neg v)$ by the same amount, until the weight of one of the sets becomes zero; note that if ϕ is true, $T(v)$ and $T(\neg v)$ must be disjoint. That is, setting $\delta = \min_{b \in \{0,1\}} w(T(v \leftarrow b))$, we decrease the weights of variables in each of $T(v \leftarrow b)$ and $T(v \leftarrow \neg b)$ by δ in an arbitrary way that does not decrease any of the variable weights below zero. (When v is a variable, we use $T(v \leftarrow 1)$ to denote $T(v)$, and $T(v \leftarrow 0)$ to denote $T(\neg v)$.) Note that the total weight of variables has been reduced by $2 \cdot \delta$; this is where the approximation factor of 2 arises. After the reduction of weights has been performed, there is some value b for which $w(T(v \leftarrow b)) = 0$, and we set v to such a b.

The analysis Note that if the formula ϕ is satisfiable, the strategy Σ_0 playing against any adversary always produces an outcome satisfying all constraints, as it never makes a move that could falsify the formula. The approximation guarantee of Σ_0 is expressed in the following theorem.

Theorem 3. *Let Φ be a weighted bijunctive quantified formula, and assume that Φ is satisfiable. Then $\mathsf{cost}(\Phi, \Sigma_0) \leq 2 \cdot \mathsf{cost}(\Phi)$.*

Proof. We shall prove the theorem by induction on the number of variables, and the sum of the variable weights. We consider three cases.

- $\Phi = \forall y \ldots$
  In this case we have

$$\begin{aligned}\mathsf{cost}(\Phi, \Sigma_0) &\leq \max_b \mathsf{cost}(\Phi[y \leftarrow b], \Sigma_0)\\ &\leq \max_b 2 \cdot \mathsf{cost}(\Phi[y \leftarrow b])\\ &\leq 2 \cdot \mathsf{cost}(\Phi).\end{aligned}$$

  Note that when $b = 1$, we assume the cost of $\Phi[y \leftarrow b]$ to include $w(y)$, the weight of y.
- $\Phi = \exists x \ldots$, $\Phi[x \leftarrow b]$ is true, and $\Phi[x \leftarrow \neg b]$ is false (where $b \in \{0, 1\}$). In this case we have

$$\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi[x \leftarrow b], \Sigma_0) \leq 2 \cdot \mathsf{cost}(\Phi[x \leftarrow b]) = 2 \cdot \mathsf{cost}(\Phi).$$

- $\Phi = \exists x \ldots$, and $\Phi[x \leftarrow b]$ is true for both $b \in \{0, 1\}$.
  There are two subcases.
  If $\delta = \min_{b \in \{0,1\}} w(T(x \leftarrow b)) > 0$, then Σ_0 decreases the weight of variables in each of $T(x \leftarrow 0)$ and $T(x \leftarrow 1)$ by δ. Let Φ' denote the formula with decreased weights. Observe that $\mathsf{cost}(\Phi') + \delta \leq \mathsf{cost}(\Phi)$, since every strategy must pay for setting to true the variables either in $T(x \leftarrow 0)$ or $T(x \leftarrow 1)$. Observe also that $\mathsf{cost}(\Phi, \Sigma_0) \leq \mathsf{cost}(\Phi', \Sigma_0) + 2 \cdot \delta$, because the total decrease in weights from Φ to Φ' is $2 \cdot \delta$. From these two facts it follows that

$$\mathsf{cost}(\Phi, \Sigma_0) \leq \mathsf{cost}(\Phi', \Sigma_0) + 2 \cdot \delta \leq 2 \cdot \mathsf{cost}(\Phi') + 2 \cdot \delta \leq 2 \cdot \mathsf{cost}(\Phi).$$

  If $w(T(x \leftarrow b)) = 0$ for some b, then Σ_0 sets $x \leftarrow b$. In this case we claim

$$\mathsf{cost}(\Phi[x \leftarrow b]) \leq \mathsf{cost}(\Phi),$$

  and hence that Σ_0 has made an optimal move. From the induction hypothesis, the claim implies

$$\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi[x \leftarrow b], \Sigma_0) \leq 2 \cdot \mathsf{cost}(\Phi[x \leftarrow b]) \leq 2 \cdot \mathsf{cost}(\Phi).$$

  To justify the claim, take any existential winning strategy Σ which sets $x \leftarrow \neg b$. Based on Σ we can define a new existential winning strategy Σ' that sets all literals in $T(x \leftarrow b)$ to true, and otherwise behaves just like Σ. Let Π' be the worst ignorant adversary for Σ'; it can be verified that $\mathsf{cost}(\Phi, \Sigma', \Pi') \leq \mathsf{cost}(\Phi, \Sigma, \Pi')$, from which it follows that $\mathsf{cost}(\Phi, \Sigma') \leq \mathsf{cost}(\Phi, \Sigma)$. □

4.2 The implicative hitting set case

We shall assume that each constraint of a k-IHS-B+ formula is either a disjunction of up to k variables, or an implication $u \rightarrow v$, where u and v are variables. We call a constraint of the former type a *set clause*. We define the implication graph $G(\phi)$ of such a formula ϕ to have one vertex for each variable of ϕ, and a directed edge $u \to v$ for each constraint $u \Rightarrow v$. Note that, in contrast to the implication graph of a bijunctive formula, the implication graph of a k-IHS-B+ formula does not contain any vertices for negative literals. The closure $T(v)$ of a vertex v is of $G(\phi)$ is the set of vertices reachable from v via a directed path; the vertex v is considered to be reachable from itself.

Before describing our strategy, we define a normal form for an implicative hitting set formula. We say that a quantified IHS-B+ formula ϕ is in *normal form*, if the following three conditions hold.

- There is no directed path from a universally quantified variable v_i to another universal variable v_j in the graph $G(\phi)$. Otherwise, the adversary could play $v_i = 1$, $v_j = 0$, hence falsifying at least one constraint on the path form v_i to v_j.
- When v_i is an existential variable, and v_j is an universal variable, such that v_1 comes before v_j in the quantification order, there is not a directed path from v_i to v_j, nor a directed path from v_j to v_i.
  To justify this assumption, suppose there is a path from v_i to v_j. Any winning existential strategy must set $v_i = 0$, since otherwise the universal player could falsify some constraint on the path by playing $v_j = 0$. Similarly, if there is a path from v_j to v_i the value of v_i is forced to 1. Hence we can replace v_i with the appropriate value and propagate the changes.
- The implication graph $G(\phi)$ is acyclic. If $G(\phi)$ is not acyclic, consider some strongly connected component C. Let v_i be the variable in C that occurs first in the quantification order. Note that the only way how to satisfy all constraints of C is to play $v_j = v_i$ for all variables $v_j \in C$ (note that all variables in C, with the possible exception of v_i, are by our previous assumptions existential). Hence we can replace all occurences of v_j by v_i and set the weight of v_i to be $w(C)$.

Throughout, we assume IHS-B+ formulas to be in normal form. We have the following characterization of satisfiable IHS-B+ formulas.

Theorem 4. *Let ϕ be a IHS-B+ quantified formula in normal form. Let E be the set of all its existential variables v, such that $T(v)$ contains no universal variable. Then Φ is satisfiable if and only if each set clause of Φ contains a variable from E.*

Proof. If each set clause of Φ contains a variable from E, the existential player can win by playing $v \leftarrow 1$ for all $v \in E$, and $v \leftarrow 0$ for $v \notin E$. If the condition of the lemma is not met, there must be a set clause C that contains no variable in E. By setting all universal variables to 0, the universal player can force the existential player to either violate an implicative constraint, or set all variables outside E to 0, violating the clause C. $\square$

The strategy Our strategy, denoted by Σ_0, is constructed for a given formula Φ in a recursive fashion. In particular, we show how for a formula Φ to construct a simpler formula Φ', and, given a strategy for Φ', construct a strategy for the original formula Φ. The strategies we construct are ignorant: each existential variable has its value predetermined, and is set independently of the universal player's moves. Clearly, such a strategy can be described in space polynomial in the size of the original formula Φ. It can easily be verified that all of the reduction steps we will give can be performed in polynomial time.

The simplification process proceeds as follows. If Φ contains no set clauses, the optimal strategy for the universal player is to set all of the universal variables to 1. Our strategy in this case is to set all variables v such that $T(v)$ contains a universal variable to 1, and all other variables to 0.

If Φ contains a set clause, we do the following. Pick an arbitrary set clause $u_1 \vee u_2 \vee \ldots \vee u_l$, and decrease the weight of all sets $T(u_i)$ in such a way that the weight in each set is decreased by at least the amount $\delta = \min_i w(T(u_i))$, and the total decrease in the weight of all variables is no more than $l\delta$. (This can be done, for example, by considering the sets $T(u_i)$ in any order, and decreasing the weight of variables in each set until either the weight of the variables in the set is decreased by δ, or the weights of all variables in the set become zero.) Note that after the weight reduction, there must be a variable u_i such that $w(T(u_i)) = 0$. There are two cases: if $T(u_i)$ contains only existential variables, we set all variables in $T(u_i)$ to 1. Otherwise, $T(u_i)$ must contain a universal variable y such that $w(T(y)) = 0$. In this case we can assume that the universal player plays $y \leftarrow 0$. Hence, we set $y \leftarrow 0$, and set $u \leftarrow 0$ for all existential variables u such that $y \in T(u)$, including the existential variable u_i.

The analysis As in the previous case, it is straightforward to verify that the described strategy Σ_0 is winning on satisfiable formulas. We establish the claimed approximation ratio in the following theorem.

Theorem 5. *Let Φ be a weighted k-IHS-B+ quantified formula, and assume that Φ is satisfiable. Then* $\mathsf{cost}(\Phi, \Sigma_0) \leq k \cdot \mathsf{cost}(\Phi)$.

Proof. We consider three cases.

- If Φ contains no set clauses, our strategy Σ_0 sets $v \leftarrow 1$ if $T(v)$ contains a universal variable and $v \leftarrow 0$ otherwise. We claim that $\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi)$. This is because against an adversary that sets all universal variables to 1, every winning strategy must set to 1 all variables that Σ_0 sets to 1.
- If Φ contains a set clause $C = u_1 \vee \ldots \vee u_l$ such that $\delta = \min_i w(T(u_i)) > 0$, our algorithm constructs a new formula Φ', where the weights of the sets $T(u_i)$ have been decreased by δ. Since Φ' is a simpler formula, we can assume by induction that $\mathsf{cost}(\Phi', \Sigma_0) \leq k \cdot \mathsf{cost}(\Phi')$. Observe that $\mathsf{cost}(\Phi') + \delta \leq \mathsf{cost}(\Phi)$, because any winning existential strategy must ensure that one of the variables in C is set to 1. Observe also that $\mathsf{cost}(\Phi, \Sigma_0) \leq \mathsf{cost}(\Phi', \Sigma_0) + k \cdot \delta$, because the total decrease in weights from Φ to Φ' is no more than $k\delta$. From these two facts it follows that

$$\mathsf{cost}(\Phi, \Sigma_0) \leq \mathsf{cost}(\Phi', \Sigma_0) + k \cdot \delta \leq k \cdot \mathsf{cost}(\Phi') + k \cdot \delta \leq k \cdot \mathsf{cost}(\Phi).$$

- If Φ contains a set clause $C = u_1 \vee \ldots \vee u_l$ such that $w(T(u_i)) = 0$ for some i, there are two subcases.
  If $T(u_i)$ does not contain any universal variables, our strategy Σ_0 sets all variables in $T(u_i)$ to 1. Let $\Phi' = \Phi[T(u_i) \leftarrow 1]$ be the simplified formula. ¿From induction we get $\mathsf{cost}(\Phi', \Sigma_0) \leq k \cdot \mathsf{cost}(\Phi')$. By definition of Σ_0, it holds that $\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi', \Sigma_0)$. It also holds that $\mathsf{cost}(\Phi') \leq \mathsf{cost}(\Phi)$. This is because any strategy Σ can be modified into one that sets all variables in $T(u_i)$ to 1, and has value less than or equal to the value of Σ. Hence, $\mathsf{cost}(\Phi, \Sigma_0) \leq k \cdot \mathsf{cost}(\Phi)$.
  On the other hand, if $T(u_i)$ contains an universal variable y, our strategy Σ_0 sets y and all variables v such that $y \in T(v)$ to 0. Again, let Φ' be the formula with these substitutions made. We have $\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi', \Sigma_0)$ by definition of Σ_0. It is straightforward to verify that $\mathsf{cost}(\Phi') \leq \mathsf{cost}(\Phi)$, so we have $\mathsf{cost}(\Phi, \Sigma_0) = \mathsf{cost}(\Phi', \Sigma_0) \leq k \cdot \mathsf{cost}(\Phi') \leq k \cdot \mathsf{cost}(\Phi)$, as desired.

4.3 The weakly negative case

The strategy In this case, the strategy Σ_0 is quite simple. As in the bijunctive case, as the game progresses and variables are set, Σ_0 instantiates the variables with the values played. When the formula has the form $\phi = \exists x \ldots$ and it is time for the existential player to set a variable x, the strategy simply checks to see if the formulas $\phi[x \leftarrow 0]$, $\phi[x \leftarrow 1]$ are satisfiable. If only one is, the strategy sets x accordingly; otherwise, the strategy simply sets x to 0. Note that determining satisfiability of a weakly negative quantified formula can be performed in polynomial time [7].

The analysis We now demonstrate that the described strategy is optimal in the sense that it yields a 1-approximation.

Theorem 6. *Let Φ be a weighted weakly negative quantified formula, and assume that Φ is satisfiable. Then $\mathsf{cost}(\Phi, \Sigma_0) \leq \mathsf{cost}(\Phi)$.*

Proof. Without loss of generality, we focus on weakly negative quantified formulas of the form $\phi = \forall y_1 \exists x_1 \ldots \forall y_n \exists x_n \mathcal{C}$. Let $\phi[a_1, b_1, \ldots, a_k, b_k]$ denote the formula

$$\forall y_{k+1} \exists x_{k+1} \ldots \forall y_n \exists x_n \mathcal{C}[y_1 = a_1, x_1 = b_1, \ldots, y_k = a_k, x_k = b_k],$$

that is, the formula where the variables $y_1, x_1, \ldots, y_k, x_k$ are instantiated with the values $a_1, b_1, \ldots, a_k, b_k \in \{0, 1\}$. We make the observation that if the formulas $\varphi = \phi[a_1, b_1, \ldots, a_k, b_k]$ and $\varphi' = \phi[a'_1, b'_1, \ldots, a'_k, b'_k]$ are both satisfying, then $\varphi'' = \phi[a_1 \wedge a'_1, b_1 \wedge b'_1, \ldots, a_k \wedge a'_k, b_k \wedge b'_k]$ is also satisfying: given a winning strategy $\Sigma = \{\sigma_i\}$ for φ and a winning strategy $\Sigma' = \{\sigma'_i\}$ for φ', it can be verified using [5, Lemma 4.8] that $\{\sigma_i \wedge \sigma'_i\}$ is a winning strategy for φ''.

Fix a quantified formula ϕ and let Σ' be an arbitrary winning strategy for ϕ. To demonstrate that Σ_0 is a 1-approximation, it suffices to show the following:

for any ignorant strategy Π, and for all existentially quantified variables x in ϕ, it holds that $A_{(\phi,\Sigma_0,\Pi)}(x) \leq A_{(\phi,\Sigma',\Pi)}(x)$. (Here, $\leq$ denotes the usual total ordering on $\{0,1\}$ where $0 \leq 1$.) Suppose that this is not true, that is, there is an ignorant strategy Π and an existentially quantified variable x such that $A_{(\phi,\Sigma_0,\Pi)}(x) = 1$ and $A_{(\phi,\Sigma',\Pi)}(x) = 0$. Let x_i denote the first variable with this property, and define $a_j = A_{(\phi,\Sigma_0,\Pi)}(v)$ and $a'_j = A_{(\phi,\Sigma',\Pi)}(v)$ for $j \in [i-1]$. Moreover, let b_j denote the value given to y_j by Π (for $j \in [n]$). We have that the formulas $\phi[a_1, b_1, \ldots, b_{k-1}, a_k, 1]$ and $\phi[a_1, b'_1, \ldots, b'_{k-1}, a_k, 0]$ are satisfiable, from which it follows by the initial observation that $\phi[a_1, b_1, \ldots, b_{k-1}, a_k, 0]$ is also satisfiable, contradicting the definition of Σ_0. $\square$

5 Inapproximability

In this section, we establish the inapproximability of the remaining constraint languages. Note that the class of width-2 affine constraints is contained in the class of bijunctive constraints.

Theorem 7. *If S is a constraint language that is not weakly negative, bijunctive, nor k-IHS-B+, then* MIN-QCSP(S) *is not constant factor approximable, unless* NP *has deterministic algorithms running in time $n^{\log^{O(1)} n}$.*

Proof. As in [3], we define $[S]$ to be the class of all relations definable from S using relations from $S \cup \{=\}$ (where $=$ is the equality relation on $\{0,1\}$), conjunction, existential quantification, and universal quantification; we refer the reader to [3] for more details. Note that if S is weakly negative (bijunctive, k-IHS-B+), then $[S]$ is also weakly negative (respectively bijunctive, k-IHS-B+); see [5, Lemma 4.5] for a part of the proof.

As in [3, Proposition 1], it can be demonstrated that for two constraint languages S_1, S_2, if $[S_1] \subseteq [S_2]$, then MIN-QCSP(S_1) is polynomial-time reducible to MIN-QCSP(S_2). ¿From this, it follows that MIN-QCSP(S) and MIN-QCSP$([S])$ are polynomial-time reducible to each other.

Let S be a constraint language satisfying the hypothesis of the theorem. Then, $[S]$ is not weakly negative, bijunctive, nor k-IHS-B+. We make use of the Pol and Inv operators as defined (for instance) in [4], as well as the description and notation of Post's lattice in [2]. It can be verified that either Pol$([S]) \subseteq S_0$, Pol$([S]) \subseteq V$, or Pol$([S]) \subseteq L$. Let MIN-CSP(S) denote the restriction of the problem MIN-QCSP(S) to instances where there is no universal quantification.[2] We break into three cases depending on which of these containments holds, and show that in any of these cases, MIN-CSP(T) can be reduced to MIN-QCSP(S) for some T that is not weakly negative, bijunctive, k-IHS-B+, nor 0-valid.

- If Pol$([S]) \subseteq S_0$, then $[S]$ is not 0-valid. We have that MIN-CSP$([S])$ reduces to MIN-QCSP(S).

[2] Note that MIN-CSP(S) is denoted by MIN-ONES(S) in [8].

– If $\mathsf{Pol}([S]) \subseteq V$, then the relation $R = \{0,1\}^3 \setminus \{(0,0,1\}$ is in $[S]$. It follows by the definition of $[S]$ that the relation $R' = \{0,1\}^2 \setminus \{(0,0)\}$ is also in $[S]$, implying that $[S]$ is not 0-valid. We have that $\mathsf{MIN\text{-}CSP}([S])$ reduces to $\mathsf{MIN\text{-}QCSP}(S)$.
– If $\mathsf{Pol}([S]) \subseteq L$, then we assume that $[S]$ is 0-valid (for if it is not, we can simply use the reduction from $\mathsf{MIN\text{-}CSP}([S])$ to $\mathsf{MIN\text{-}QCSP}(S)$). We claim that $\mathsf{MIN\text{-}CSP}([S] \cup \{N\})$ reduces to $\mathsf{MIN\text{-}QCSP}([S])$, where $N = \{(1)\}$. We first observe that $M = \{(0,0),(1,1)\} \in [S]$. Given an instance of $\mathsf{MIN\text{-}CSP}([S] \cup \{N\})$, we create an instance of $\mathsf{MIN\text{-}QCSP}([S])$ having the same cost by introducing a single universally quantified variable y (which comes before all of the existential variables) and converting each constraint $N(v)$ over N to the constraint $M(y,v)$. Thus $\mathsf{MIN\text{-}CSP}([S] \cup \{N\})$ reduces to $\mathsf{MIN\text{-}QCSP}([S])$, which in turn reduces to $\mathsf{MIN\text{-}QCSP}(S)$.

¿From the classification theorem for MIN-CSP along with the associated hardness of approximation results in [8], it follows that $\mathsf{MIN\text{-}CSP}(T)$, and hence $\mathsf{MIN\text{-}QCSP}(S)$, is not constant factor approximable, unless NP has deterministic algorithms running in time $n^{\log^{O(1)} n}$. $\square$

References

1. B. Aspvall, M.F. Plass, R.E. Tarjan. A linear time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, vol 8 no. 3, 1979.
2. E. Böhler, N. Creignou, S. Reith and H. Vollmer. Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory. *ACM SIGACT-Newsletter*, 34(4):38-52, 2003.
3. Ferdinand Börner, Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Quantified Constraints: Algorithms and Complexity. *Computer Science Logic*, 2003.
4. Andrei Bulatov, Andrei Krokhin, Peter Jeavons. Constraint satisfaction problems and finite algebras. *Proceedings of ICALP'00*, Lecture Notes in Computer Science 1853, 272-282.
5. Nadia Creignou, Sanjeev Khanna, Madhu Sudan. Complexity Classifications of Boolean Constraint Satisfaction Problems. *SIAM Monographs on Discrete Mathematics and Applications*, 2001.
6. D. Gusfield, L. Pitt. A Bounded Approximation for the Minimum Cost 2-Sat Problem. *Algorithmica*, 8:103–117, 1992.
7. Marek Karpinski, Hans Kleine Bning, Peter H. Schmitt. On the Computational Complexity of Quantified Horn Clauses. *Computer Science Logic*, 1987, 129-137.
8. S. Khanna, M. Sudan, L. Trevisan and D.P. Williamson. The Approximability of Constraint Satisfaction Problems. *SIAM Journal on Computing*, 30:1863–1920, 2001.
9. Thomas J. Schaefer. The complexity of satisfiability problems. *Proceedings of the Tenth ACM Symposium on Theory of Computing*, pp. 216–226, 1978.