Quantifiers and Approximation

Alessandro Panconesi
Desh Ranjan*

TR 89-1061
November 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

*Supported by NSF research grant DCR-85-20597.

Quantifiers and Approximation

Alessandro Panconesi and Desh Ranjan *
Department of Computer Science
Cornell University, Ithaca, NY 14853

November 26, 1989

Abstract

We investigate the relationship between logical expressibility of NP optimization problems and their approximation properties. First such attempt was made by Papadimitriou and Yannakakis, who defined the class of NPO problems MAX NP. We show that many important optimization problems do not belong to MAX NP and that in fact there are problems in P which are not in MAXNP. The problems we consider fit naturally in a new complexity class that we call MAX Π_1. We prove that several natural optimization problems are complete for MAX Π_1 under approximation preserving reductions. All these complete problems are non approximable unless P = NP. This motivates the definition of subclasses of MAX Π_1 that only contain problems which are presumably easier with respect to approximation. In particular, the class that we call RMAX(2), contains approximable problems and problems like MAX CLIQUE that are not known to be non-approximable. We prove that MAX CLIQUE and other natural optimization problems are complete for RMAX(2). All the complete problems in RMAX(2) share the interesting property that they either are non-approximable or are approximable to any degree of accuracy.

*Supported by NSF research grant DCR 85-20597

1 Introduction

Approximation of NP optimization (NPO) problems is an important area in the theory of algorithms [5,12]. Although there is a wealth of results providing ingenious algorithms for approximation of individual problems, and several isolated proofs of non-approximability of others (assuming $P \neq NP$), there is a lack of unifying theoretical framework and the reasons why a problem enjoys particular approximation properties are not clear [1,3,2,5,10,11,13,2].

In order to develop a theory for approximation of NPO problems, one has to define subclasses of NPO with problems in the same subclass having similar approximation properties. Defining these classes in terms of Turing machines presents a fundamental problem; changing something "computationally insignificant" like the value of a single bit, can have enormous effect on the approximation properties of the computed function.

Krentel, in [9], develops a theory of NPO problems where the complexity of a problem is measured by the number of queries that any P^{SAT} machine, computing the optimization function for the problem, makes to its SAT oracle. The results are elegant, but are not related to approximation; for example, MAX KNAPSACK and MIN TSP are both complete for class of functions in P^{SAT}, but while the first is approximable in a very strong sense, the second is not approximable unless P = NP.

In [3], subclasses of NPO are defined in terms of Turing machines. The results are interesting, but it seems doubtful that meaningful problems can be proven complete in those classes.

To avoid the problems which arise in the Turing machine model, Papadimitriou and Yannakakis [13] introduce an approach based on the logical characterization of NP given by Fagin [4]; this result states that NP is the set of languages that are the generalized spectrum of a second order existential formula, ranging over finite structures. They use this characterization to define a natural class of NPO problems, which they call MAX NP. Roughly speaking, a problem in MAX NP has the property that the set of its feasible solutions can be described by a formula of the type $\exists \bar{y}\ \Phi(\bar{y}, S)$, where Φ is quantifier-free, S is a feasible solution and $\bar{y}$ ranges over the input structure, such as a graph or a boolean formula. Interestingly enough, they show that all the problems in MAX NP are approximable and that there is a *uniform* fashion in which they can be approximated.

This raises two questions. The first is if there are approximable problems that are not in this class, that is, what is the expressive power of the class. The second, more general one, is what kind of relationship, if any, there is between the logical representation of a problem and its approximation properties.

In this paper, we first prove that the expressive power of MAX NP is rather limited. We prove that well-known and important problems like MAX CLIQUE, MAX 3DM, and MAX 3SC (optimization versions of 3DM the SET COVERING) are not in MAX NP. It is not known if MAX CLIQUE is approximable but we prove that MAX 3DM and MAX 3SC are approximable. In fact, we also prove the stronger fact that MAX NP does not capture P because we show that the problem of finding a maximum matching can not be expressed in MAX NP. These are expressibility results and do not rely on any assumptions such as P $\neq$ NP.

It turns out that all these problems which cannot be expressed as problems in MAX NP have similar logical structure and they fit nicely into a new class that we call MAX Π_1. Loosely speaking, these problems have the property that the set of feasible solutions can be described by means of a first order formula of the type $\forall \bar{y}\ \Phi(\bar{y}, S)$.

We investigate the structure of the class MAX Π_1 and find natural complete problems under reductions that preserve approximability [3,13]. For example we prove that, given a boolean formula, the problem of finding a satisfying assignment that sets to true the maximum number of variables (we call this problem MAX ONES) is MAX Π_1-complete. MAX Π_1 in its full-fledged form turns out to be too expressive; the complete problems for MAX $\Pi_1$are not approximable unless P = NP. But the problems like MAX 3DM, MAX CLIQUE, MAX 3SC do not seem to be that difficult and in fact they are either approximable or are not known to be non-approximable. We define subclasses of MAX Π_1 which still capture these problems and where the approximability of the complete problems for the class is an open question. These subclasses are defined by restricting the structure of the logical formulae allowed to express the problems. The motivation for the constraints imposed comes from observing the similarity in the expressions for the problems mentioned above. The major limitation that is imposed corresponds to saying that if S is a feasible solution for the problem and $S' \subset S$, then so is S'. The smallest and most interesting of these subclasses contains MAX 3DM, MAX 3SC, and has MAX CLIQUE and MAX GRAPH k-COLORING as complete problems. The other classes have a natural generalization of MAX CLIQUE as their complete problems. All of the complete problems share the interesting property that either they are non-approximable or are approximable for any ϵ.

This paper is organized as follows. Section 2 contains the necessary definitons. In Section 3 we prove that MAX CLIQUE, MAX 3DM, MAX 3SC, and the maximum matching problem are not in MAX NP. We then introduce the class MAX Π_1 and prove that the above problems belong to it. In Section 4 we prove the MAX Π_1-completeness of the problems MAX ONES and MAX NSF with respect to approximation preserving reductions. In Section 5, we define a subclass of MAX Π_1, the class RMAX, and prove completeness results for several optimization problems.

2 Definitions

Definition 1 *An* NPO *problem is a tuple $F = (\mathcal{I}_F, S_F, f_F, \mathrm{opt})$ where*

- $\mathcal{I}_F \subseteq \Sigma^*$ *is the space of* input instances. *It is recognizable in polynomial time.*
- $S_F(x)$ *is the space of* feasible solutions *on input $x \in \mathcal{I}_F$. We can define $S_F(x)$ as $S_F(x) = \{y \mid |x| \leq q_F(|x|) \wedge \pi_F(x, y)\}$ where q_F is a polynomial and π_F is a polynomial time computable predicate.*
- $f_F : S_F(x) \to N$, *the* objective function, *is a polynomial time computable function.*
- $\mathrm{o}pt \in \{\max, \min\}$.

As an example we briefly state how MAX CLIQUE can be expressed in the above formalism. The set of input instances is the set of all encodings of undirected graphs $G = (V, E)$ over Σ^*.

The set of feasible solutions for G is the set of all cliques contained in G. The objective function is the cardinality $\|C\|$ of a given clique. The goal is to find a clique of maximum size.

In this paper we consider only the maximization problems.

It seems possible to define NPO more concisely in the following way; $F \in$ NPO if there exists a polynomial time NDTM such that, for all $x \in \mathcal{I}_F$

$$\mathrm{opt}_F(x) = \max_y \{N(x,y) \mid N(x,y) \text{ is the value of } N(x) \text{ along the computation path } y\}.$$

However, this definition is unsatisfactory as it does not explicitely state what the set of feasible solutions and the objective function are, and we need these two objects if we want to develop a theory of approximation.

To define approximation we need to define a notion of error [5,8,12].

Definition 2 *The* relative error *of a feasible solution with respect to the optimum of an* NPO *problem F is defined as*

$$\mathcal{E}(\mathrm{opt}_F(x), y) = \frac{|\,\mathrm{opt}_F(x) - f_F(y)\,|}{\mathrm{opt}_F(x)}$$

where $y \in S_F(x)$.

Definition 3 *An* NPO *problem F is* approximable, *if there exist a polynomial time algorithm A and $\epsilon \in (0,1)$ such that, for all instances $x \in \mathcal{I}_F$:* i) *$A(x) \in S_F(x)$, and* ii) *$\mathcal{E}(\mathrm{opt}_F(x), A(x)) \leq \epsilon$.*

Definition 4 APX *is the class of all approximable* NPO *problems.*

Examples of problems in APX are MAX SAT, MAX CUT, MIN ΔTSP, MIN BIN PACKING and MIN NODE COVER [5,12,8].

It is well known that some NPO problems can be approximated within *any* ϵ (see [5,12]).

Definition 5 *An* NPO *problem is said to have a* polynomial time approximation scheme *if there exists an algorithm $A(x,\epsilon)$ such that, for all ϵ and all $x \in \mathcal{I}_F$:* i) *$A(x,\epsilon) \in S_F(x)$, and* ii) *$\mathcal{E}(\mathrm{opt}_F(x), A(x,\epsilon)) \leq \epsilon$. The complexity of A must be polynomial for any* fixed *ϵ.*

To clarify the definition, the complexity of a polynomial time approximation scheme can be something like $2^{1/\epsilon}p(|\,x\,|)$ or $|\,x\,|^{1/\epsilon}$; these cases actually arise in practice (see [7,5]).

Definition 6 PTAS *is the class of* NPO *problems that have a polynomial time approximation scheme.*

Examples of problems in this class are some scheduling problems [7], and MAX KNAPSACK (that actually enjoys even stronger approximation properties [12]).

It follows directly from the definitions that PTAS $\subseteq$ APX $\subseteq$ NPO. It is not difficult to see that these inclusions are strict if P $\neq$ NP. One of the most important open problems in the area is to understand under what conditions a problem is in APX or in PTAS. A variety of problems can be proved to be in APX but it is not known if they are in PTAS. Some examples are MAX SAT, MAX CUT, MIN ΔTSP, and MIN NODE COVER.

In a recent paper [13], Papadimitriou and Yannakakis have addressed this question. Instead of defining NPO problems in terms of Turing machines, they use a logical characterization of NP due to Fagin [4]; a language is in NP if and only if it is the generalized spectrum of a second order existential formula, ranging over finite structures.

For example,

$$\varphi \in \text{SAT} \Leftrightarrow \exists T\ \forall c\ \exists x\ (P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x))$$

Intuitively, T is a second-order variable that ranges over truth assignments; φ is described by means of the two binary predicates P and N; $P(x,c) = \text{TRUE}$ iff variable x appears positive in clause c. Similarly, $N(x,c) = \text{TRUE}$ iff variable x appears negated in clause c. The formula $(P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x))$ ensures that T sets to true at least one literal for each clause.

In general, for any language L in NP there is a *quantifier free* formula Φ_L such that

$$I \in L \Leftrightarrow \exists S\ \forall \overline{x}\ \exists \overline{y}\ \Phi_L(I, S, \overline{x}, \overline{y})$$

Informally, the instance I is described with a finite structure $I = \{A, P_1^{a_1}, .., P_k^{a_k}\}$, where A is a finite set and $P_i^{a_i} \subseteq A^{a_i}$ a predicate of arity a_i. In the formula Φ_L, I stands for the set of predicates $P_i^{a_i}$ (this is an abuse of notation, for more formal description see [4]). $S \subseteq A^s$ is a predicate of arity s describing the solution (e.g a satisfying assignment), and $\overline{x}, \overline{y}$ are vectors of fixed arity of elements of A. We could consider a more general format where S too is a collection of predicates; in this paper we consider the case where S is a single predicate for sake of simplicity, but our proofs generalize to the general case.

It is important to realize the the formula Φ_L is the same for all instances I. In particular it is of fixed size, and the arities of the vectors $\overline{x}, \overline{y}$, together with the arities of the predicates appearing in I and S are fixed.

This formalism can be used to express NPO problems too. Again, for sake of clarity, we consider an example. Take the problem MAX SAT: given a boolean formula φ in *CNF* , find an assignment that maximizes the number of clauses set to true.

Then, for all instances φ the following holds

$$opt_{\text{MAX SAT}}(\varphi) = \max_T \| \{c \mid \exists x\ (P(x,c) \wedge T(x)) \vee (N(x,c) \wedge \neg T(x))\} \|.$$

Definition 7 *[13]* MAX NP *is the class of* NPO *problems* F *such that*

$$opt_F(I) = \max_S \| \{\overline{x} \mid \exists \overline{y}\ \Phi_L(I, S, \overline{x}, \overline{y})\} \|$$

where Φ_L *is a quantifier free formula.*

Theorem 1 *[13] Every problem in* MAX NP *can be approximated within some fixed ratio, i.e.* MAX NP $\subseteq$ APX.

This theorem is an indication that there might be a close connection between the logical representation of a problem and its approximation properties.

To address the question of which MAX NP problems are in PTAS, in [13] is introduced a suitable approximation preserving reducibility. We use the more general definition of reducibility used in [3].

Definition 8 *Given two* NPO *problems Fand G, a* PTAS preserving reduction *(P-reduction) from F to G is a triple $f = (t_1, t_2, c)$ such that*

i) $t_1 : \mathcal{I}_F \to \mathcal{I}_G$ *and* $t_2 : \mathcal{I}_F \times S_G(t_1(x)) \to S_G(x)$ *are polynomially computable, and* $c : (0,1) \to (0,1)$ *is computable.*

ii) f *is such that, for all* $x \in \mathcal{I}_F$ *and for all* $\epsilon \in (0,1)$

$$\mathcal{E}(\mathrm{opt}_G(t_1(x)), y) \leq c(\epsilon) \;\Rightarrow\; \mathcal{E}(\mathrm{opt}_F(x), t_2(x,y)) \leq \epsilon.$$

Most of the reductions in this paper will actually be a much stronger form of reduction.

Definition 9 *A P-reduction from F to G is said to be an* approximation preserving reduction *(A-reduction) if* $c(\epsilon) = \epsilon$.

In a P-reduction, we use t_1 to map instances of F into instances of G, and t_2 to map approximated solution for G back into approximated solutions of F. The relation among t_1, t_2 and c ensures that the following propositions hold.

Proposition 1 *If* $G \in$ PTAS *and* $F \leq_P G$, *then* $F \in$ PTAS.

Proposition 2 *P-reductions compose.*

In [13] it is shown that several natural problems are MAX NP-complete under P-reductions. From Proposition 1 it follows that

Proposition 3 *If a* MAX NP*-hard problems is in* PTAS *then* MAX NP $\subseteq$ PTAS.

In their paper, Papadimitriou and Yannakakis have left it as an open question if there are approximable optimization problems that are not in MAX NP . We answer this question positively in the next section.

3 Expressiveness of MAX NP

In this section we show that a a whole range of NPO problems are not in MAX NP. These include polynomial time computable problems like the problem of finding the maximum matching in a graph, approximable problems like MAX 3DM and MAX 3SC (optimization versions of 3DM and SET COVERING), and problems like MAX CLIQUE whose approximability is an open question. All these problems naturally belong to a new complexity class that we call MAX Π_1.

The following theorem is an instance of a more general principle

$$\mathcal{A} \models \exists x\, \varphi(x) \;\wedge\; \mathcal{A} \subseteq \mathcal{B} \;\Rightarrow\; \mathcal{B} \models \exists x\, \varphi(x)$$

where $\varphi(x)$ is quantifier-free.

Theorem 2 MAX CLIQUE $\notin$ MAX NP.

PROOF. For simplicity, we make the assumption that the representation of a graph is the usual one, i.e. a graph is a finite structure $G = (V, E)$ where V is the set of vertices and E is the edge predicate.

The proof is by contradiction. Assume that MAX CLIQUE belongs to MAX NP, therefore we know that for all G's

$$opt_{CLQ}(G) = \max_S \| \{\overline{x} \mid \exists \overline{y}\ \varphi(\overline{x}, \overline{y}, E, S)\} \|$$

where $\bar{x} = (x_1, ..., x_t)$, $\overline{y} = (y_1, ..., y_r)$, and $S \subseteq V^s$. Let us consider one particular $G_1 = (V_1, E_1)$ (with the only requirement that it has non empty edge set) and let S_1 be such that

$$opt_{CLQ}(G_1) \;=\; \| \{\overline{x} \mid \exists \overline{y}\ \varphi(\overline{x}, \overline{y}, E_1, S_1)\} \|$$

Now we construct a new graph G_{new}. Let $G_2 = (V_2, E_2)$ be an isomorphic copy of G_1 and let $G_{new} = (V_1 \cup V_2, E_1 \cup E_2)$, i.e. there are no edges between G_1 and G_2. Hence,

$$opt_{CLQ}(G_{new}) = opt_{CLQ}(G_1) = opt_{CLQ}(G_2) \stackrel{def}{=} \text{OLDVALUE}$$

We claim that,

$$\max_S \| \{\overline{x} \mid \exists \overline{y}\ \varphi(\overline{x}, \overline{y}, E_{new}, S)\} \| \;\geq\; 2\ \text{OLDVALUE}$$

Given a tuple $\overline{x}$ of vertices in G_1 we will indicate with $\overline{x}'$ the tuple made of the isomorphic elements in G_2. Similarly, S_2 is the "isomorphic copy" of S_1 that is

$$< a_1, ..., a_s > \in S_1 \;\Leftrightarrow\; < a_1', ..., a_s' > \in S_2$$

We have that

$$G_1, S_1 \models \exists \overline{y}\ \varphi(\overline{a}, \overline{y}, E, S) \;\Leftrightarrow\; G_2, S_2 \models \exists \overline{y}\ \varphi(\overline{a}', \overline{y}, E, S)$$

Choose $S_{new} = S_1 \cup S_2$. We claim that, for all $\bar{a} \in V_1^t$,

$$G_1, S_1 \models \exists \overline{y}\ \varphi(\overline{a}, \overline{y}, E, S) \;\Rightarrow\; G_{new}, S_{new} \models \exists \overline{y}\ \varphi(\overline{a}, \overline{y}, E, S).$$

To see this, assume $\bar{a}$ be such that

$$G_1, S_1 \models \exists \overline{y}\ \varphi(\overline{a}, \overline{y}, E, S).$$

This implies that there exists $\overline{b}$ such that $\varphi(\overline{a}, \overline{b}, E_1, S_1)$ is true. We will show that $\varphi(\overline{a}, \overline{b}, E_{new}, S_{new})$ is also true. To this end, we will show that the truth values of the atoms of φ in the two cases are the same. The atoms of $\varphi(\overline{a}, \overline{b}, E_{new}, S_{new})$ are of the form $E_{new}(\bar{z})$, $S_{new}(\overline{w})$, where $\bar{z}$ and $\overline{w}$ are tuples of elements taken from the set $\{a_1, ..., a_t, b_1, ..., b_r\}$, or $x = y$ where x and y range over $\{a_1, ..., a_t, b_1, ..., b_r\}$. But then, since $S_{new} = S_1 \cup S_2$ and $E_{new} = E_1 \cup E_2$,

$$E_{new}(\overline{w}) = E_1(\overline{w}) \quad \text{and} \quad S_{new}(\bar{z}) = S_1(\bar{z}).$$

As a consequence,

$$\varphi(\overline{a},\overline{b},E_1,S_1) = \varphi(\overline{a},\overline{b},E_{new},S_{new}).$$

Analogously, $\exists\overline{y}\ \varphi(\overline{x}',\overline{y},E_2,S_2) \Rightarrow \exists\overline{y}\ \varphi(\overline{x}',\overline{y},E_{new},S_{new})$, and hence

$$\|\{\overline{x} \mid \exists\overline{y}\ \varphi(\overline{x},\overline{y},E_{new},S_{new})\}\| \geq 2\ \text{OLDVALUE}$$

because $\{\bar{x} \mid \exists\overline{y}\ \varphi(\overline{x},\overline{y},E_1,S_1)\}$ and $\{\bar{x} \mid \exists\overline{y}\ \varphi(\overline{x}',\overline{y},E_2,S_2)\}$ are disjoint. □

The theorem was proved under the assumption that a graph is a finite sructure of the kind $G = (V, E)$. However, what we actually used in the proof were the following assumptions on the coding of graphs via finite structures. First, isomorphic graphs are represented by isomorphic structures and isomorphic structures represent isomorphic graphs. Second, if $G_1 = (V_1, E_1)$ has a coding $G_1 = (A_1, P_1^1, .., P_m^1)$ and $G_2 = (V_2, E_2)$ has a coding $G_2 = (A_2, P_1^2, .., P_m^2)$ then $G = (V_1 \cup V_2, E_1 \cup E_2)$ has coding isomorphic to $G = (A_1 \cup A_2, P_1^1 \cup P_1^2, .., P_n^1 \cup P_n^2)$.

We now introduce two optimization problems. We first show that they are approximable and then that they do not belong to MAX NP.

The first problem is the optimization version of 3DM and it's a natural generalization of the maximum matching problem.

Suppose we are given a set of 3-tuples $T = \{T_1, .., T_n\}$. A set $M \subseteq T$ is a *matching* if it is a collection of mutually component-wise different tuples: $A = (a_1, a_2, a_3) \in M \ \wedge\ B = (b_1, b_2, b_3) \in M \ \Rightarrow\ a_i \neq b_i,\ 1 \leq i \leq 3$). We then define the following NPO problem.

MAX 3 DIMENSIONAL MATCHING (MAX 3DM).

INSTANCE. A collection of 3-tuples $T = \{T_1, .., T_n\}$.

PROBLEM. Find the maximum size matching.

In a similar fashion, we can define the problem MAX KDM where the input is a collection of k-tuples.

Theorem 3 MAX KDM *in is* APX.

PROOF. One can show that the size of any *maximal* matching is at least $1/k$ of the size of a maximum matching. □

Theorem 4 MAX 3DM *is not in* MAX NP.

PROOF. The proof is similar to that of Theorem 2. Assume, by contradiction, that MAX 3DM $\in$ MAX NP. Then, there exists a formula φ such that for all instances I of the problem

$$opt_{3DM}(I) = \max_S \| \{\overline{x} \mid \exists \overline{y}\ \varphi(\overline{x}, \overline{y}, I, S)\} \|.$$

Consider an instance $I_1 = \{T_1, .., T_n\}$ such that $opt_{3DM}(I_1) = n$. Given $T_i = (a_i, b_i, c_i)$ and $T_j = (a_j, b_j, c_j)$, we say that they are *compatible* if $a_i \neq a_j \ \wedge \ b_i \neq b_j \ \wedge \ c_i \neq c_j$. I_1 is a set of n mutually compatible 3-tuples.

From our contradictory assumption we have that there is S_1 such that

$$opt_{3DM}(I_1) \ = \ \| \{\overline{x} \mid \exists \overline{y}\ \varphi(\overline{x}, \overline{y}, I_1, S_1)\} \| = n.$$

Let $\overline{x}_1, .., \overline{x}_n$ be the tuples satisfying the above formula. Take $\overline{x}_1$ and suppose, without loss of generality, that it contains a_1, i.e. $\overline{x}_1 = (a_1, u_2, .., u_k)$.

We now construct another instance I_2 by simply replacing a_1 with a brand new element a_0. Let $I_2 = \{T_0, T_2, .., T_n\}$ where $T_0 = (a_0, b_1, c_1)$. I_2 is made of the same tuples of I_1 except the first, T_0. T_0 and T_1 only differ for the first component. We choose a_0 so that I_2 is made of n mutually compatible tuples. Now define S_2 to be the same set as S_1 provided any occurence of a_1 is replaced by an occurrence of a_0, and define $\overline{z}_i$ to be the same tuple as $\overline{x}_i$ provided the same substitution takes place. Then,

$$\| \{\overline{z} \mid \exists \overline{y}\ \varphi(\overline{z}, \overline{y}, I_2, S_2)\} \| = n$$

If we now consider the new instance $I_{new} = I_1 \cup I_2$ and define $S_{new} = S_1 \cup S_2$, we have that $opt_{3DM}(I_{new}) = n$ but

$$\max_S \| \{\overline{w} \mid \exists \overline{y}\ \varphi(\overline{w}, \overline{y}, I_{new}, S)\} \| \ \geq \ \| \{\overline{w} \mid \exists \overline{y}\ \varphi(\overline{w}, \overline{y}, I_{new}, S_{new})\} \| \ \geq \ n+1$$

because

$$\| \{\overline{x}_1, .., \overline{x}_n\} \ \cup \ \{\overline{z}_1, .., \overline{z}_n\} \| \geq n+1.$$

□

Basically the same proof applies to MAX KDM.

Corollary 1 *For all k,* MAX KDM *does not belong to* MAX NP.

An important thing to be noticed is that this corollary is true even when $k = 2$, i.e. when the problem is just finding the maximum matching in a graph. It is well known that this problem is in P; it follows that MAX NP does not include all polynomially computable optimization problems.

We now introduce another problem, similar to MAX KDM. Given a collection of sets of cardinality k, $S = \{S_1, .., S_n\}$, we define a *covering* $C \subseteq S$ to be a collection of mutually disjoint sets: $S_i \in C \ \wedge \ S_j \in C \ \Rightarrow \ S_i \cap S_j = \emptyset$.

MAX K-SET COVERING (MAX KSC).

INSTANCE. A collection of sets of cardinality k, $S = \{S_1, .., S_n\}$.

PROBLEM. Find the covering of maximum size.

We claim, without proof, that the following theorems hold. Their proof is very similar to the theorems we saw for MAX KDM.

Theorem 5 *For all k,* MAX KSC *is in* APX.

Theorem 6 *For all k,* MAX KDM *does not belong to* MAX NP.

All the problems we introduced in this section fit nicely in a new complexity class.

Definition 10 MAX Π_1 *is the class of* NP *optimization problems F such that, for all input instances I,*

$$\mathrm{opt}_F(I) = \max_S \| \{\overline{x} \mid \forall \overline{y}\, \varphi(G, S, \overline{x}, \overline{y})\} \|$$

As an example, consider MAX CLIQUE. It is easy to see that, for all graphs G,

$$\mathrm{opt}_{CLQ}(G) = \max_C | \{x \mid C(x) \ \wedge \ \forall yz\ (C(y) \wedge C(z) \to E(y,z) \vee x = y)\} |$$

where x, y, and z range over vertices and $E(y,z) =$ TRUE iff $(y,z) \in E$.
We state the next proposition omitting the proof.

Proposition 4 MAX CLIQUE, MAX KDM, MAX KSC *belong to* MAX Π_1.

MAX Π_1 is a natural way of expressing many NPO problems. In the next section we will prove completeness for natural variants of SAT.
In particular, our canonical complete problem will be the following.

MAX NUMBER OF ONES (MAX ONES).

INSTANCE. A boolean formula φ in *3CNF*.

PROBLEM. Find a satisfying assignment with the maximum number of variables set to true.

We can express MAX ONES as a MAX Π_1 problem as follows. As in the case of 3SAT the instance is coded by means of 4 predicates $C_0, .., C_3$ where $C_i(x,y,z) =$ TRUE iff φ has a clause where the first i among its three variables x, y, and z appear negated. Then,

$$\mathrm{opt}_{ONES}(\varphi) = \max_T \| \{x \mid T(x) \wedge \forall yzw\ \Phi(\varphi, T, x, y, z, w)\} \|$$

where

$$\begin{aligned} \Phi(\varphi, T, x, y, z, w) \ = \ & (C_0(y,z,w) \to T(y) \vee T(z) \vee T(w)) \wedge \\ & (C_1(y,z,w) \to \neg T(y) \vee T(z) \vee T(w)) \wedge \\ & (C_2(y,z,w) \to \neg T(y) \vee \neg T(z) \vee T(w)) \wedge \\ & (C_3(y,z,w) \to \neg T(y) \vee \neg T(z) \vee \neg T(w)) \end{aligned}$$

4 Structural Properties of MAX Π_1

In this section we exhibit complete problems for the class MAX Π_1 . We also show that the complete problems for the class are non-approximable unless P = NP. Our first MAX Π_1-complete problem is the following.

MAX NUMBER OF SATISFIABLE FORMULAE (MAX NSF).

> INSTANCE. A set of *3CNF* formulae $\{\varphi_1, \varphi_2, ..., \varphi_n\}$.
>
> PROBLEM. Find a truth assignment to the variables such that the maximum number of the formulae are satisfied.

In this problem, the set of feasible solutions of non zero weight are the assignments satisfying at least one formula φ_i; this implies that approximating MAX NSF is NP-hard.

Theorem 7 MAX NSF *is* MAX Π_1*-complete under P-reductions.*

PROOF. MAX NSF $\in$ MAX Π_1 as the optimum value on instance I can be expressed as

$$opt_{\text{MAX NSF}}(I) = \max_T \| \{ i \mid \varphi_i(T) = \text{TRUE},\ 1 \le i \le n \} \|$$

where I is the input instance $\{\varphi_1, \varphi_2, ..., \varphi_n\}$ and T a unary predicate which is basically a truth assignment to the variables in $\{\varphi_1, \varphi_2, ..., \varphi_n\}$.

We assume that the input, I, is presented via two 3-ary predicates P and N where $P(i,j,k)$ is true iff variable x_k occurs positively in the jth clause of the formula φ_i , C_{ij}, and $N(i,j,k)$ is true iff variable x_k occurs negatively in C_{ij}. Then, more precisely,

$$opt_{\text{MAX NSF}}(I) = \max_T \| \{ i \mid \forall j k_1 k_2 k_3\ \Phi(I, T, i, j, k_1, k_2, k_3) \} \|$$

where

$$\begin{aligned}
\Phi \;=\; & (P(i,j,k_1) \wedge P(i,j,k_2) \wedge P(i,j,k_3) \rightarrow T(k_1) \vee T(k_2) \vee T(k_3)) \;\wedge \\
& (P(i,j,k_1) \wedge P(i,j,k_2) \wedge N(i,j,k_3) \rightarrow T(k_1) \vee T(k_2) \vee \neg T(k_3)) \;\wedge \\
& (P(i,j,k_1) \wedge N(i,j,k_2) \wedge N(i,j,k_3) \rightarrow T(k_1) \vee \neg T(k_2) \vee \neg T(k_3)) \;\wedge \\
& (N(i,j,k_1) \wedge N(i,j,k_2) \wedge N(i,j,k_3) \rightarrow \neg T(k_1) \vee \neg T(k_2) \vee \neg T(k_3)).
\end{aligned}$$

Now we establish the completeness of MAX NSF .

Let F be any optimization problem in MAX Π_1, and let f_F be its optimization function. Then

$$opt_F(I) = max_S \| \{ \bar{x} \mid \forall \bar{y}\ \Psi(\bar{x}, \bar{y}, I, S) \} \|.$$

Recall that $\bar{x}, \bar{y}$ represent tuples of variables of fixed arity. Hence, each tuple ranges over a polynomially sized domain (in the size of I). Then,

$$\{\bar{x} \mid \forall \bar{y}\ \Psi(\bar{x}, \bar{y}, I, S)\} = \{\bar{x}_i \mid \forall \bar{y} \Psi(\bar{x}_i, \bar{y}, I, S)\ \bar{x}_i \in domain(\bar{x})\}.$$

Now let $\varphi_i = \bigwedge_j \Psi(\bar{x}_i, \bar{y}_j, I, S)$. Each φ_i is a polynomially sized boolean formula whose variables are $S(v_1, .., v_l)$ where S is an l-ary predicate. Moreover there are exactly $\|domain(\bar{x})\|$ of these formulae. Observe that, since Ψ is a formula of fixed length it can be put in *CNF* and, with the introduction of new variables, can be changed into a *3CNF* formula maintaining satisfiability (or unsatisfiability). Hence, we can assume that each φ_i is a *3CNF* formula.

Now observe that, for any predicate assignment S_0 to S, the corresponding truth assignment S' to $\{S(v_1, ..v_l) \mid (v_1, ..v_l) \in domain(S)\}$ given by

$$S(v_1, \ldots v_l) = \text{TRUE} \Leftrightarrow (v_1, \ldots v_l) \in S_0$$

makes k formulae φ_i true iff $f_F(S_0) = k$.

This is an A-reduction. We can conclude that MAX NSF is complete for MAX Π_1 under P-reductions. □

We now show the MAX Π_1-completeness of MAX ONES with respect to P-reductions. We have already shown at the end of section 3 that MAX ONES is in MAX Π_1. To show hardness, we first exhibit a reduction from MAX NSF into an intermediate problem, MAX DONES, and then reduce MAX DONES to MAX ONES. MAX DONES is the following problem.

MAX DISTINGUISHED ONES (MAX DONES).

INSTANCE. A boolean formula $\varphi(X, Z)$ where $X = \{x_1, .., x_n\}$ and $Z = \{z_1, .., z_n\}$. The z_i's are the *distinguished* variables.

PROBLEM. Find a satisfying assignment for φ with the maximum number of distinguished variables set to true.

Lemma 1 MAX DONES *is* MAX Π_1*-complete with respect to P-reductions.*

PROOF. MAX DONES can be written down as a MAX Π_1 problem in essentially the same way we wrote MAX ONES; besides the predicates C_i we need a predicate $D(z)$ that is TRUE iff z is a distinguished variable.

We now reduce MAX NSF to MAX DONES. Given an instance $\psi = \{\varphi_1(X), .., \varphi_n(X)\}$ of MAX NSF we construct the formula

$$F(X, Z) = (\varphi_1 \vee \neg z_1) \wedge \ldots \wedge (\varphi_n \vee \neg z_n)$$

By distributing the z_i's, we can see that F is a 4CNF formula such that $z_{i_1}, .., z_{i_k}$ are TRUE iff $\varphi_{i_1}, .., \varphi_{i_k}$ are TRUE.

This is an A-reduction. To complete the proof we have to transform $F(X, Z)$ into a *3CNF* formula. This can be done by introducing extra undistinguished variables y_i's; a clause $(x_1 \vee$

$x_2 \vee x_3 \vee x_4)$ is mapped into the two clauses $(x_1 \vee x_2 \vee y_1) \wedge (\neg y_1 \vee x_3 \vee x_4)$. Since the y's are non-distinguished, this is again an A-reduction. □

Theorem 8 MAX ONES *is* MAX Π_1*-complete.*

PROOF. We have already established that MAX ONES $\in$ MAX Π_1 at the end of the preceding section. To prove completeness, we transform MAX DONES into MAX ONES. Let $\varphi(Z, X)$ with $Z = \{z_1, ..., z_p\}$ and $X = \{x_1, .., x_q\}$ be the instance of MAX DONES; we transform it into an instance $\psi(X, Y, Z, Z')$ of MAX ONES. In what follows we will indicate with $\tau' : X \cup Y \cup Z \cup Z' \to \{0, 1\}$ a satisfying assignemnt for ψ, and with τ the restriction to $X \cup Z$ of τ'. The reduction we are going to show is such that $\psi(\tau'(X), \tau'(Y), \tau'(Z), \tau'(Z')) = \text{TRUE} \Leftrightarrow \varphi(\tau(X, Z)) = \varphi(\tau'(X), \tau'(Z)) = \text{TRUE}$.

The instance of MAX ONES is the following formula

$$\psi(X, Y, Z, Z') = \varphi(Z, X) \wedge \beta(Z, Z') \wedge \alpha(X, Y, Z).$$

Y and Z' are sets of brand new variables while X and Z are the same variables appearing in φ.

In ψ any true variable contributes to the weight of a satyisfying assignment τ'. We would like the contribution of the x's and y's to be negligible with respect to that of the z's. The mission of the subformula $\beta(Z, Z')$ is to privilege the variables z's. We define

$$\beta(Z, Z') = \bigwedge_{1 \leq i \leq p} \left(\bigwedge_{1 \leq j \leq 2l-1} z_i \equiv z_{ij} \right)$$

where $Z = \{z_1, .., z_p\}$ and $Z' = \{z_{ij} \mid 1 \leq i \leq q,\ 1 \leq j \leq 2l - 1\}$. The index l is set to $l = q + r$, where r is the number of y's in ψ. l is selected after the construction of α is done. What β does is equivalent to assigning a weight of $2l$ to each z_i. Notice that β can be expressed in *CNF* with clauses of two literals. Also notice, that any satisfying assignment for $\varphi(X, Z)$ automatically determines a satisfying assignments for $\beta(Z, Z')$.

The mission of $\alpha(X, Y, Z)$ is to forbid truth assignments of ψ where some of the x_i's are set to true and all the z_i's are set to false. If this happened, we would have a solution of ψ with cost greater than zero mapped into a solution of φ of cost zero, i.e. approximated solutions would not be mapped into approximated solutions (recall that our transformation simply considers restrictions $\tau(X, Z) = \tau'(X, Z)$, where τ' satisfies ψ).

A way of implementing α would be to write down

$$\alpha = \bigwedge_{1 \leq i \leq q} \left(\neg x_i \vee \bigvee_{1 \leq j \leq q} z_j \right).$$

But these are clauses of unbounded length. To have clauses of length three, we transform each clause $(\neg x_i \vee z_1 \vee .. \vee z_q)$ into

$$(\neg x_i \vee z_1 \vee y_1) \wedge (\neg y_1 \vee z_2 \vee y_2) \wedge \ldots \wedge (\neg y_{q-1} \vee z_q).$$

It can be checked that $\alpha(X,Y,Z)$ so defined satisfies the two properties: *i)* the truth of any y_i or x_i implies the truth of some z_j; *ii)* any truth assignment τ' of $\varphi(Z,X)$ can be extended to a truth assignment $\tau \supseteq \tau'$ satisfying $\alpha(X,Y,Z)$.

To summarize, $\psi(X,Y,Z,Z')$ can be expressed in *3CNF* and the restrictions to $X \cup Z$ of its satisfying assignments form the set of satyisfying assignments for $\varphi(X,Z)$.

We now have to show that the reduction is an P-reduction. The transformation can certainly be carried over in polynomial time.

Let $opt_{DONES}(\varphi) = k$, by construction it follows that $opt_{ONES}(\psi) \geq 2lk$. It also follows that the possible weights for a solution τ' of ψ are $w(\tau') = 0, 2l + n_1, .., 2li + n_i, ..2lk + n_k$ where $1 \leq i \leq k$ and $n_i \leq l = q + r$ for all i's. Moreover, the relationship between a solution τ' and its restriction τ is $w(\tau') = 2li + n_i \Leftrightarrow w(\tau) = i$ and $w(\tau') = 0 \Leftrightarrow w(\tau) = 0$.

We want to show

$$\frac{opt(\psi) - w(\tau')}{opt(\psi)} \leq \frac{\epsilon}{2} \Rightarrow \frac{opt(\varphi) - w(\tau)}{opt(\varphi)} \leq \epsilon$$

In order to do so, it is enough to prove

$$\frac{opt(\psi) - w(\tau')}{opt(\psi)} \geq \frac{opt(\varphi) - w(\tau)}{2\ opt(\varphi)}$$

Consider a solution τ' of ψ. When $w(\tau') = 0$ or $w(\tau') \geq 2lk$ the above equation holds. Suppose then that $w(\tau') = 2li + n_i$ with $1 \leq i \leq k - 1$. We have

$$\begin{aligned}
\frac{\epsilon}{2} &\geq \frac{opt(\psi) - w(\tau')}{opt(\psi)} \\
&\geq \frac{2lk - w(\tau')}{2lk} \qquad (1) \\
&\geq \frac{2lk - l(2i+1)}{2lk} \qquad (2) \\
&= \frac{2k - (2i+1)}{2k} \\
&\geq \frac{k-i}{2k} \\
&= \frac{opt(\varphi) - w(\tau)}{2\ opt(\varphi)}
\end{aligned}$$

Equation (1) holds since $opt(\psi) \geq 2lk$, while equation (2) holds since $w(\tau') = 2li + n_i \leq l(2i+1)$. This concludes the proof. $\square$

The complete problems we saw are non approximable unless P = NP. However, we know that MAX Π_1 contains approximable problems and *self improvable* problems like MAX CLIQUE, which are either non approximable or are in PTAS. It would be interesting to characterize classes of approximable problems inside MAX Π_1. In the next section, we see how it is possible to describe problems like MAX CLIQUE, MAX KSC, and MAX KDM by posing syntactic restrictions on the formulae Φ_F certifying membership of F in MAX Π_1.

5 Restrictions on Expressive Power of MAX Π_1

In the previous section we saw that MAX Π_1 in its full generality has problems which are too hard for approximation. On the other hand, let us examine the expressions for the optimization functions for various problems we have been discussing, and which we proved are not in MAX NP

- MAX CLIQUE.

$$opt_{CLQ}(G) = max_C \|\{x \mid C(x) \wedge \forall yz\ (y = z \vee \neg C(y) \vee \neg C(z) \vee E(y,z))\}\|$$

- MAX 3DM.

$$opt_{3DM}(I) = max_M \|\{\bar{a} \mid\ M(\bar{a}) \wedge \forall \bar{b}\bar{c}\ \Phi(M,T,\bar{b},\bar{c})\}\|$$

where

$$\phi(M,T,\bar{b},\bar{c}) = (\neg M(\bar{b}) \vee T(\bar{b})) \wedge \left((\bar{b} = \bar{c}) \vee \neg M(\bar{b}) \vee \neg M(\bar{c}) \vee \bigwedge_{i=1,2,3} (b_i \neq c_i)\right).$$

Here $\bar{a}$ stands for (a_1, a_2, a_3) and I stands for the input instance (A, T) where $T \subseteq A^3$ with $T(\bar{a}) =$ TRUE iff *"$\bar{a}$ is a triple"*.

These problems are not only in MAX Π_1, but the fashion in which they are expressed is also rather similar. More precisely, all these problems can be expressed as:

$$opt_F(I) = \max_S\{\| S \| \mid \forall \bar{y}\ \Phi_F(\bar{y}, I, S)\}$$

where $\| S \|$ denotes $\| \{\bar{x} \mid S(\bar{x})\} \|$, $\bar{y}$ is a first-order variable and Φ_F is quantifier-free. Most importantly, *if Φ_F is expressed in CNF then all occurrences of S occur negatively.*

Definition 11 *A problem $F \in$ RMAX(K) if its optimization function can be expressed as*

$$opt_F(I) = \max_S\{\|S\| \mid \forall \bar{y}\ \Phi(\bar{y}, I, S)\}$$

where Φ is a quantifier-free CNF formula with all occurrences of S in Φ being negative, S a single predicate appearing at most k times in each clause, and $\|S\|$ denotes $\|\{\bar{x}|S(\bar{x})\}\|$.

Definition 12 RMAX $= \bigcup_k$ RMAX(K).

This subclass may seem very restricted in the beginning, but it captures many of the problems in MAX Π_1 which are provably not in MAX NP. In fact, most of the problems we have considered are in RMAX(2) . Other problems which fall into this class include :

- MAX SC: this is a generalization of MAX KSC. Given a collection $S_1, \ldots, S_n$ of finite sets, find a covering of maximum size. Notice, MAX SC = $\bigcup_k$ MAX KSC. This problem and MAX KSC are in RMAX(2). Similarly, MAX DM and MAX KDM are in RMAX(2).

- MAX INDEPENDENT SET: given a graph, find the size of the maximum independent set. This problem is in RMAX(2).

- MAX GRAPH K-COLORING: given a graph $G = (V, E)$ and an integer k, find the maximum number of vertices of G that can be colored with k colors such that no two adjacent vertices have the same color. This problem is in RMAX(2).

- MAX k-ANLSAT: this is the restriction of MAX ONES where all the variables appear negated in the input formula, and where every clause has at most k literals. This problem is in RMAX(K).

- MAX K-HYPERCLIQUE: An input instance is a k-hypergraph $H = (A, E)$ where A is a set and $E \subset \mathcal{P}(A)$ and $e \in E \Rightarrow 1 \leq |e| \leq k$. An element of E is called a hyperedge. A feasible solution is any set $W \subset A$ satisfying $\{u_1, \ldots, u_i\} \subseteq W,\ i \leq k \ \Rightarrow\ \{u_1, \ldots, u_i\} \in E$. Such a set is called a *k-hyperclique*. The goal is to find a k-hyperclique of maximum size.

  This problem is a generalization of the CLIQUE problem for graphs to hypergraphs and it is a trivial fact that MAX CLIQUE $\equiv_P$ MAX 2-HYPERCLIQUE.

Thus there is a large class of problems which are in RMAX(K). We now establish two theorems. The first theorem is about the equivalence of the families of problems MAX K-HYPERCLIQUE and MAX k-ANLSAT. The second theorem shows that any problem in RMAX(K) reduces to one of the problems in the above families.

Theorem 9 $\forall k$ MAX K-HYPERCLIQUE $\equiv_P$ MAX k-ANLSAT.

PROOF. (Sketch) We first show that MAX K-HYPERCLIQUE$\leq_P$ MAX k-ANLSAT. Let $H = (A, E)$ be any k-hypergraph. Construct ϕ_H as follows. The variables of ϕ_H are $\{x_i | i \in A\}$. ϕ_H is a conjunction of all the clauses of the form $(\neg x_{i_1} \vee \neg x_{i_2}, \ldots \vee \neg x_{i_k})$ where $x_{i_j} \in var(\phi_H)$ and $\{i_1, i_2, \ldots i_k\}$ is not a hyperedge in E. The literals may be repeated within a clause in which case it is simplified. These are the only clauses of ϕ_H. It can now be checked that ϕ_H can be satisfied with $x_{i_1}, x_{i_2}, \ldots x_{i_l}$ all set to TRUE if and only if $\{i_1, i_2, \ldots i_l\}$ form a hyperclique in H.

To prove that MAX k-ANLSAT $\leq_P$ MAX K-HYPERCLIQUE, use the inverse mapping. $\square$

Theorem 10 $F \in$ RMAX $\Rightarrow$ $\exists k$ $F \leq_L$ MAX k-ANLSAT.

PROOF. Let F be any problem in RMAX. Then the optimization function of F can be expressed as :

$$opt_F(I) = \max_S\{||S|| \mid \forall \bar{y}\ \Phi_F(\bar{y}, I, S)\}$$

where Φ_F is in *CNF* and all occurrences of S in Φ_F are negated. Since Φ_F is a fixed formula, there is a constant k such that Φ_F is in k- *CNF* . We then claim that F reduces to MAX k-ANLSAT under P-reductions.

Consider boolean formula $\phi_{F,I}$ with variables $S(\bar{x})$.

$$\phi_{F,I} = \bigwedge_j \Phi_F(\bar{y}_j, I, S)$$

with $\bar{y}_j$ ranging over domain($\bar{y}$), which is polynomial in the size of the input. Just to stress, the only variables in $\phi_{F,I}$ are $S(\bar{x})$ because each of the atomic formulae involving the predicates describing the input becomes TRUE or FALSE on assigning a value to $\bar{y}$ depending upon the input I. By the assumption on the structure of Φ_F all occurrences of $S(\bar{x})$ occur negated in $\phi_{F,I}$. Hence it is an instance of MAX k-ANLSAT.

Now if $\phi_{F,I}$ is satisfied with $S(\bar{x}_1), S(\bar{x}_2)\ldots S(\bar{x}_l)$ set to TRUE then choosing the predicate assignment S_0 for S where $S_0(\bar{x})$ = TRUE iff $\bar{x} \in \{\bar{x}_1, \bar{x}_2 \ldots \bar{x}_l\}$ satisfies $\forall\bar{y}\Phi_F(\bar{y}, I, S_0)$ and $||S_0|| = l$. Moreover, if $opt_F(I) = M$, and this is witnessed by S_0 with $\bar{x}_1, \bar{x}_2 \ldots \bar{x}_M \in S_0$ then $\phi_{F,I}$ can be satisfied with M variables $S(\bar{x}_1), S(\bar{x}_2), \ldots S(\bar{x}_M)$ set to TRUE.

This establishes the result. □

The last theorem has interesting consequences.

Theorem 11 *The problems* MAX CLIQUE, MAX GRAPH k-COLORING, MAX SC, *and* MAX DM *are* RMAX(2)*-complete with respect to A-reductions.*

PROOF. (Sketch) The completeness of MAX CLIQUE follows from Theorem 10 and the trivial fact that MAX CLIQUE $\equiv_A$ MAX 2-HYPERCLIQUE. The remaining reductions are easy to obtain; for example, MAX CLIQUE $\leq_A$ MAX GRAPH k-COLORING because MAX GRAPH 1-COLORING is the same as MAX INDEPENDENT SET. □

All the RMAX(2)-complete problems share a very interesting property: either they are non approximable or they are in PTAS.

Definition 13 *A problem $F \in$ NPO is* self improvable *if there is a P-reduction from F to itself such that:* i) $\mathrm{opt}_F(t_1(x)) = (\mathrm{opt}_F(x))^2$; ii) *if $y \in S_F(t_1(x))$ is a feasible solution of weight k then $t_2(x, y) \in S_F(x)$ is a feasible solution of weight at least $\sqrt{k}$.*

This definition could be made more general.

If a problem is self-improvable, then it is either in NPO − APX or in PTAS. The reason is that we can apply the reduction n times to map x into $t_1(x)^n$; an error of ϵ in $t_1(x)^n$ corresponds to an error ϵ_n in x, and when n tends to infinity, ϵ_n tends to 0. MAX CLIQUE is self-improvable [12,6].

Fact 1 *If F A-reduces to G and G is self-improvable, so is F.*

We then have the following corollary.

Corollary 2 *All the complete problems of Theorem 11 are self-improvable.*

Notice that these results are obtained without directly mapping these problems to themselves.

6 Conclusion

We have investigated the relationship between the logical expressibility of NPO problems and their approximation properties. To summarize, we have first shown that class MAX NP is rather weak in its expressive power. We have then defined another class of NPO problems based on logical structure. For this class we have demonstrated complete problems; moreover we have obtained interesting subclasses where the complete problems have similar properties with respect to approximation and in addition they all have the property of self-improvability. This work is a step in the direction of developing a general framework for estabilishing a connection between logical structure of a problem and its approximation properties and we hope that it provides an impetus for the same.

Acknowledgements

The first author would like to thank Pierluigi Crescenzi and Riccardo Silvestri; informal discussions with them were the starting point of this work. We thank Dexter Kozen for leading us to the proof of Theorem 2. We are also indebted to Radha, Tushar and Steve Vavasis for insightful and humorous criticism. Last, but not the least, the second author gratefully acknowledges the support of his advisor Professor Juris Hartmanis.

References

[1] G.Ausiello, A. Marchetti Spaccamela and M. Protasi *"Toward a unified approach for the classification of NP-complete problems"* Theor Comput Sci, **12** (1980), 83-96

[2] D.Bruschi, D.Joseph and P.Young *"A structural Overview of NP Optimization Problems"* Comp Sci Tech Report 861, Univ of Wisconsin-Madison (1989)

[3] P.Crescenzi and A.Panconesi *"Completeness in Approximation Classes"*, Proc FCT '89, Lectures Notes 380, Springer-Verlag, 116-126 (1989)

[4] R.Fagin *"Generalized First-Order Spectra, and Polynomial-Time Recognizable Sets"* in Complexity and Computations, R. Karp editor, AMS, (1974)

[5] M.R.Garey and D.Johnson *"Computers and Intractability: a guide to the theory of NP-completeness"* Freeman, San Francisco 1979

[6] M.R.Garey and D.Johnson *"The complexity of Near-Optimal Graph Coloring"* J ACM **23** (1976), 43-49

[7] D.Hochbaum and D.Shmoys *"Using Dual Approximation Algorithms for Scheduling problems: theoretical and practical results"* J ACM **34** (1987),

[8] D.Johnson *"Approximation algorithms for combinatorial problems"* J Computer and System Sci, **9** (1974), 256-278

[9] M.W.Krentel *"The Complexity of Optimization Problems"* J Computer and System Sci, **36** (1988), 490-509

[10] P.Orponen and H.Mannila *"On approximation preserving reductions: complete problems and robust measures"* Tech Report, University of Helsinki 1987

[11] A.Paz and S.Moran *"NP-optimization problems and their approximation"* Theor Comput Sci, **15** (1981), 251-277

[12] C.Papadimitriou and K.Steiglitz *"Combinatorial Optimization: Algorithms and Complexity"* Prentice-Hall, Englewood Cliffs, New Jersey 1982

[13] C.Papadimitriou and M.Yannakakis *"Optimization, Approximation, and Complexity Classes"* Proc 20th ACM Symp Theory of Comp, (1988), 229-234