Logic and Approximation

Von der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Diplom–Mathematiker
(Universität Basel)
Anders Malmström

aus

Basel

Referent: Universitätsprofessor Dr. E. Grädel
Korreferent: Universitätsprofessor Dr. J. Flum
Tag der mündlichen Prüfung: 18. Februar 1997

"D 82 (Diss. RWTH Aachen)"

Contents

Chapter 1

Introduction

Classifying optimization problems into *easy* and *hard to solve* problems is a challenging task that has been studied for many years. A problem is said to be *easy* if it is solvable by a *polynomial-time algorithm*. An algorithm is *polynomial-time*, or *efficient*, if there is a polynomial $p(n)$ such that on an input of size n the algorithm always returns a solution in time less than $p(n)$. On the other hand, we say that a problem is *hard* if no efficient algorithm is known. The principal notion of the theory that unifies these hard problems is NP-*completeness*. The class of NP-complete problems has two very interesting properties:

(i) No efficient algorithm is known that solves an NP-complete problem.

(ii) If there is an efficient algorithm for *one* NP-complete problem, then for *all* NP-complete problems there is an efficient algorithm.

These properties are the basis for the famous conjecture that P≠NP. It states that there is *no* polynomial-time algorithm that solves an NP-complete problem.

The theory of NP-completeness was introduced in terms of decision problems (problems with the answer *yes* or *no*) but its development has been influenced very much by the study of combinatorial optimization problems. Most of the natural and famous NP-complete decision problems are derived from optimization problems by putting a bound on the cost function. For example the TRAVELING SALESMAN PROBLEM looks for the shortest tour visiting n cities. The associated decision problem asks: Is there a tour of length shorter than k?
Optimization problems whose decision versions are in NP are called NP *optimization problems.*

The NP-completeness of a decision problem implies that its associated optimization problem is hard to solve. More precisely, assuming P≠NP, it rules out the possibility of finding in polynomial-time an optimal solution. This does not exclude the existence of *approximation algorithms*. An approximation algorithm is a polynomial-time algorithm that returns for every input a nearly optimal solution.

However, with respect to approximability NP optimization problems may behave differently. Some problems like MAX KNAPSACK admit approximation algorithms returning solutions up to any desired accuracy, others like the TRAVELING SALESMAN PROBLEM are as hard to approximate as to solve exactly[1]. What are the structural reasons for these differences?

A first break-through in this question was made when Papadimitriou and Yannakakis [36] discovered in some important cases a relationship between approximation properties of optimization problems and their representations by logical formulae. Motivated by Fagin's characterization of NP in terms of existential second-order logic on finite structures [14], they defined two classes of optimization problems, MAX SNP and MAX NP, and proved that for each problem $\mathcal{Q}$ in these classes there exists a polynomial-time algorithm that approximates the optimum with a constant relative error. Many natural maximization problems like MAX CUT and MAX SAT are contained in MAX NP.

Building on this work Kolaitis and Thakur [25] investigated the logical definability of optimization problems in a systematic way. Let $\mathcal{Q}$ be an optimization problem whose input instances are finite structures $\mathfrak{A}$ of a fixed signature σ, and let $\psi(\bar{S}, \bar{x})$ be a first-order formula where $\bar{S}$ is a sequence of relation variables not contained in σ.
We say that ψ defines $\mathcal{Q}$ if the cost of an optimal solution of $\mathcal{Q}$ on input $\mathfrak{A}$ is definable by the expression

$$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{A}) = \max_{\bar{S}} |\{\bar{x} : \mathfrak{A} \models \psi(\bar{S}, \bar{x})\}|.$$

Recall that Σ_n (resp. Π_n) is the class of first-order formulae in prenex normal form with n alternating blocks of quantifiers beginning with $\exists$ (resp. $\forall$). Kolaitis and Thakur defined MAX Σ_n (resp. MAX Π_n) to be the class of all maximization problems definable by Σ_n-formulae (resp. Π_n-formulae). The class MAX FO consists of those maximization problems definable in the same way by arbitrary first-order formulae. Kolaitis and Thakur proved that these classes collapse to a strict hierarchy of four levels and that all maximization problems with polynomially bounded cost functions fall into this hierarchy.

Papadimitriou and Yannakakis defined the classes MAX SNP and MAX NP to be the closures of MAX Σ_0 and MAX Σ_1 under appropriate approximation-preserving reductions (so-called L-reductions). Thus, the class MAX Σ_1 provides a sufficient *syntactic criterion for approximability*.

Papadimitriou and Yannakakis also exhibited a number of complete problems for MAX SNP under L-reductions. The question whether these complete problems can be efficiently approximated with *arbitrary* accuracy has been answered in the negative. Arora et al. [2] proved that MAX SNP-hard problems cannot be approximated with arbitrarily small error (i.e. they have no PTAS), unless P=NP.

[1] For a broad list of optimization problems and their approximation properties see [9].

Moreover Khanna et al. [24] showed that all constant approximable optimization problems can be L-reduced to a problem in Max SNP.

For minimization problems the definability classes Min Σ_n and Min Π_n are defined analogously. Surprisingly the situation is not dual to maximization. The minimization classes collapse to a strict hierarchy of only two levels and all polynomially bounded minimization problems belong to Min FO. Moreover, even the class at the bottom level, Min Σ_0, contains problems that are not efficiently approximable unless P=NP. Thus, in the case of minimization the quantifier structure of the formulae does not suffice to establish a syntactic criterion for approximability.

However, by restricting also the occurrences of the new relation symbols, Kolaitis and Thakur [25] isolated a different syntactic class, called Min $F^+\Pi_1$, all whose problems can be approximated with constant relative error. This class consists of all minimization problems whose optimum can be expressed as

$$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{A}) = \min_S \{|S| : (\mathfrak{A}, S) \models \forall \bar{x} \psi(S, \bar{x})\},$$

where S is a single relation variable and $\psi(S, \bar{x})$ is a quantifier-free formula in which all occurrences of S are positive.

In Section 3.6 we show that Min $F^+\Pi_1$ is a subclass of Min Σ_0. A canonical example of a problem in Min $F^+\Pi_1$ is Min Vertex Cover: Given a graph $G = (V, E)$, it asks for a smallest vertex cover, i.e. a set S of vertices such that every edge contains at least one vertex of S. Indeed, this problem is definable by the expression

$$\mathrm{opt}_{\mathrm{VC}}(G) = \min_S \{|S| : (G, S) \models \forall x \forall y (Exy \to (Sx \vee Sy))\}.$$

Similarly Kolaitis and Thakur [26] defined the class Min $F^+\Pi_2$. It contains an interesting subclass, Min $F^+\Pi_2(1)$, of log-approximable problems. A minimization problem is log-approximable if one can find a solution of cost at most $O(\log n)$ times larger than the cost of an optimal solution, where n is the size of the input. Examples for this class are Min Dominating Set and Min Set Cover. It is not known whether these problems can be approximated with constant relative error. In fact, this is very unlikely since Lund and Yannakakis [30] showed that there exists a constant c, such that Min Dominating Set cannot be approximated within ratio $c \log n$ unless NP is contained in DTIME$[n^{\mathrm{poly}\log n}]$.

To show that a problem is in a syntactic class it suffices to present an appropriate logical formula. Proving the opposite is more difficult. We use two different methods to prove that a specific problem does not belong to a certain class. The first is based on monotonicity arguments related to preservation theorems in logic, the second characterizes rates of growth of average optimal solutions on random inputs.

Behrendt, Compton and Grädel [3] introduced a probabilistic approach that characterizes the growth of expected optimal solution sizes for problems that belong to MAX Σ_1. The method is related to the theory of asymptotic probabilities for logically defined properties and to the probabilistic method from combinatorics. The main result states that for every problem in MAX Σ_1, the expected value of an optimal solution on a random input grows like $\Theta(p(n))$, where $p(n)$ is a polynomial and n is the cardinality of the input structure[2].

In this thesis we establish similar criteria for the classes MIN $\mathrm{F}^+\Pi_1$ and MIN $\mathrm{F}^+\Pi_2$:

Theorem.

- *Let $\mathcal{Q}$ be a problem in* MIN $\mathrm{F}^+\Pi_1$. *Then there exists a polynomial $p(n)$ such that almost surely*

  $$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{A}) = \Theta(p(n))$$

  for a randomly chosen σ-structure $\mathfrak{A}$ of size n.

- *Let $\mathcal{Q}$ be a problem in* MIN $\mathrm{F}^+\Pi_2(1)$. *Then there exist polynomials $g(n)$ and $h(n)$ such that almost surely*

  $$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{A}) = \Theta(g(n) + h(n) \cdot \log n)$$

  for a randomly chosen σ-structure $\mathfrak{A}$ of size n.

We show that on a random graph the size of a minimal dominating set grows asymptotically like $\log n$. By the theorem we can exclude membership of the MIN DOMINATING SET problem in MIN $\mathrm{F}^+\Pi_1$. Similarly, MIN GRAPH COLOURING is not in MIN $\mathrm{F}^+\Pi_2(1)$ because its optimum is almost surely $n/(2\log n)$. With the same method we can strengthen this result to the whole class MIN $\mathrm{F}^+\Pi_2$ and get that MIN GRAPH COLOURING is not in MIN $\mathrm{F}^+\Pi_2$. Hence

$$\textsc{Min } \mathrm{F}^+\Pi_1 \subsetneq \textsc{Min } \mathrm{F}^+\Pi_2 \subsetneq \textsc{Min } \mathrm{FO}.$$

This probabilistic approach was extended to definability classes of counting functions in [7].

Many natural NP optimization problems take weighted structures as input instances, e.g. graphs with weighted vertices and (or) edges. Papadimitriou and Yannakakis [36] considered variants of MAX SNP and MAX NP that use weights in a rather limited way. They attached weights to the tuples of the input structure that can be substituted for the free variables $\bar{x}$ and showed that the approximation properties are provided. In a recent paper Zimand [41] analyzed the weighted variants of all syntactically defined classes that are known to have good approximation properties.

[2] We write $f = \Theta(p(n))$ if there exist constants c and d such that $c \cdot p(n) \leqslant f(n) \leqslant d \cdot p(n)$.

To extend this definability theory to a more general one of optimization problems with arbitrary weights we use the framework of *Metafinite Model Theory* (introduced by Grädel and Gurevich [18]), an extension of finite model theory. Metafinite structures are two-sorted structures that consist of a finite structure, an infinite structure and "weight"-functions from the finite to the infinite part. In this thesis we only consider *arithmetical structures*. This are metafinite structures whose infinite part is a structure $\mathfrak{N}$ over the natural numbers.

On arithmetical structures feasible solutions of NP optimization problems can be encoded by functions from the finite part into $\mathbb{N}$. All problems we consider have in common that the cost is a linear functional in the "solution"-function (this is the case for most optimization problems).

As in the unweighted case there is a connection between the syntactic description of an optimization problem and its approximation properties.

We firstly generalize MAX Σ_1, MIN $\mathrm{F}^+\Pi_1$ and MIN $\mathrm{F}^+\Pi_2(1)$ to classes of problems operating on arithmetical structures and show that the approximation properties are preserved. As an example we get that the problem MIN k-CUT is in the generalized class MAX ${\Sigma_1}^*$ and thus constant approximable.

Secondly, we introduce a new class, MAX Φ, that contains only problems that admit a *fully polynomial-time approximation scheme* (FPTAS), i.e. problems that can be approximated to every desired accuracy ε in time polynomial in the size of the input structure and $1/\varepsilon$. MAX Φ consists of all problems that can be described in the following way:
On any given arithmetical input structure $\mathfrak{D}$

$$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{D}) = \max_f \Big\{ L(\mathfrak{D}, f) : G(\mathfrak{D}, f) \leqslant B(\mathfrak{D}) \wedge (\mathfrak{D}, f) \models \psi \Big\},$$

where L is a linear functional, G and B are numerical invariants and ψ is a first-order formula that is monotone in f. Additionally we demand that G is monotone in a very homogeneous sense, which we call *h-monotone*. For instance linear functionals are h-monotone.

A typical example for MAX Φ is the KNAPSACK problem:
The MAX 0-1 KNAPSACK problem can be described as follows. Given are n objects, $a_0, \dots, a_{n-1}$, each one with a weight $w(a_i)$ and a capacity $c(a_i)$, and an upper bound B. The goal is to choose a set of objects such that the total weight becomes maximal but the total capacity is less than B.

The input instance can be encoded as an arithmetical structure $\mathfrak{D} = (A, \mathfrak{N}, W)$ where the primary part is the pure set $A = \{a_0, a_1, \dots, a_{n-1}\}$, the secondary part is $\mathfrak{N}$ and W contains the two weight functions $w : A \to \mathbb{N}$ and $c : A \to \mathbb{N}$.
It is easy to see that MAX 0-1 KNAPSACK fits our description:

$$\begin{aligned}\mathrm{opt}_{\mathcal{Q}}(\mathfrak{D}) = \max_{f: A \to \mathbb{N}} \Big\{ \sum_{x \in A} f(x) w(x) : \sum_{x \in A} f(x) c(x) \leqslant B \\ \wedge (\mathfrak{D}, f) \models \forall x (f(x) \leqslant 1) \Big\}.\end{aligned}$$

It is well know that MAX 0-1 KNAPSACK admits an FPTAS (see [21, 5] for details). We prove the following result in Chapter 8.

Theorem. *Every problem $\mathcal{Q}$ in* MAX Φ *admits an* FPTAS.

Thus we have a syntactic description of maximization problems that induces an FPTAS. Even more, the result provides a calculus for constructing problems with an FPTAS.

We can adapt this result for minimization problems as well. We define the class MIN Φ which is the dual of MAX Φ. It consists of all minimization problems $\mathcal{Q}$ whose input instances are arithmetical structures $\mathfrak{D} = (\mathfrak{A}, \mathfrak{N}, W)$ such that

$$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{D}) = \min_{f} \Big\{ L(\mathfrak{D}, f) : G(\mathfrak{D}, f) \geqslant B(\mathfrak{D}) \wedge (\mathfrak{D}, f) \models \psi \Big\},$$

where L, G, B and ψ satisfy the same conditions as in MAX Φ. We then prove the following theorem.

Theorem. *Every problem in* MIN Φ *has an* FPTAS.

An example for MIN Φ is 2-PROCESSOR SCHEDULING.
The goal is to distribute n jobs, each one of different length $\ell(j)$, to 2 identical processors such that the total running time is minimal. Thus $A = \{a_0, \dots, a_{n-1}\}$ and

$$\mathrm{opt}_{\mathcal{Q}}(\mathfrak{D}) = \min_{f:A\to\mathbb{N}} \Big\{ \sum_{x\in A} f(x)\ell(x) : \sum_{x\in A} f(x)\ell(x) \geqslant B \wedge \forall x (f(x) \leqslant 1) \Big\},$$

where $B = \left\lceil \frac{1}{2} \sum_{x\in A} \ell(x) \right\rceil$.

Organization of the thesis:

- In Chapter 2 we introduce the basic terminology concerning NP optimization problems, approximation and on first-order logic.
- In Chapter 3 we exhibit the approach of logical representation of NP optimization problems and state relevant results, such as the hierarchy theorems. Further we show the connection between the logical description and approximation.
- In Chapter 4 we treat the separation of the logically defined classes. We introduce the two methods we use and we prove all the non-expressibility results necessary to establish the hierarchy theorems.

- In Chapter 5 we give the motivation to consider NP optimization problems that take weighted structures as input instances.
- In Chapter 6 we introduce metafinite structures and the basic terminology on metafinite logics and numerical invariants.
- In Chapter 7 we generalize the three most interesting syntactic classes, with respect to approximation, to arithmetical structures and show that the approximation properties are preserved.
- In Chapter 8 we present a syntactic description of NP optimization problems having a fully polynomial-time approximation scheme (FPTAS).