Structure of Polynomial-Time Approximation

Erik Jan van Leeuwen

Jan van Leeuwen

Technical Report UU-CS-2009-034
December 2009

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Structure of Polynomial-Time Approximation*

Erik Jan van Leeuwen
Department of Mathematics and Computer Science
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven
The Netherlands
and
Centrum Wiskunde & Informatica
PO Box 94079, 1090 GB Amsterdam
The Netherlands
E.J.van.Leeuwen@cwi.nl

Jan van Leeuwen
Department of Information and Computing Sciences
Utrecht University
Padualaan 14, 3584 CH Utrecht
The Netherlands
j.vanleeuwen@cs.uu.nl

December 28, 2009

Abstract

Approximation schemes are commonly classified as being either a polynomial-time approximation scheme (ptas) or a fully polynomial-time approximation scheme (fptas). To properly differentiate between approximation schemes for concrete problems, several subclasses have been identified: (optimum-)asymptotic schemes (ptas$^\infty$, fptas$^\infty$), efficient schemes (eptas), and size-asymptotic schemes. We explore the structure of these subclasses, their mutual relationships, and their connection to the classic approximation classes. We prove that several of the classes are in fact equivalent. Furthermore, we prove the equivalence of eptas to so-called convergent polynomial-time approximation schemes. The results are used to refine the hierarchy of polynomial-time approximation schemes considerably and demonstrate the central position of eptas among approximation schemes.

We also present two ways to bridge the hardness gap between asymptotic approximation schemes and classic approximation schemes. First, using notions from fixed-parameter complexity theory, we provide new characterizations of when problems have a ptas or fptas. Simultaneously, we prove that a large class of problems (including all MAX-SNP-complete problems) cannot have an optimum-asymptotic approximation scheme unless P=NP, thus strengthening results of Arora et al. [1]. Secondly, we distinguish a new property exhibited by many optimization problems: pumpability. With this notion, we considerably generalize several problem-specific approaches to improve the effectiveness of approximation schemes with asymptotic behavior.

1 Introduction

In the theory and practice of hard NP-optimization problems, approximation schemes are widely used for efficiently finding solutions to within any specified relative error ϵ from the optimum. Paz and Moran [38] classified these schemes into polynomial-time approximation schemes (ptas) and fully polynomial-time approximation schemes (fptas). However, the theory of approximation algorithms has led to several other useful classes of schemes, including optimum-asymptotic (ptas$^\infty$, fptas$^\infty$), efficient (eptas), and size-asymptotic (ptas$^\omega$, fptas$^\omega$) approximation schemes. It is the goal of this paper to expose the surprising connections between these seemingly unrelated notions and to study their deeper structural properties.

The foremost conclusion that follows from the results of this paper is that efficient polynomial-time approximation schemes (eptas) hold a central position in the landscape of polynomial-time approximation schemes. An eptas has a running time of the form $f(1/\epsilon)\cdot n^{O(1)}$, where n denotes the instance size and f is some computable function. The class of optimization problems admitting an

*Part of this research has been funded by the Dutch BSIK/BRICKS project.

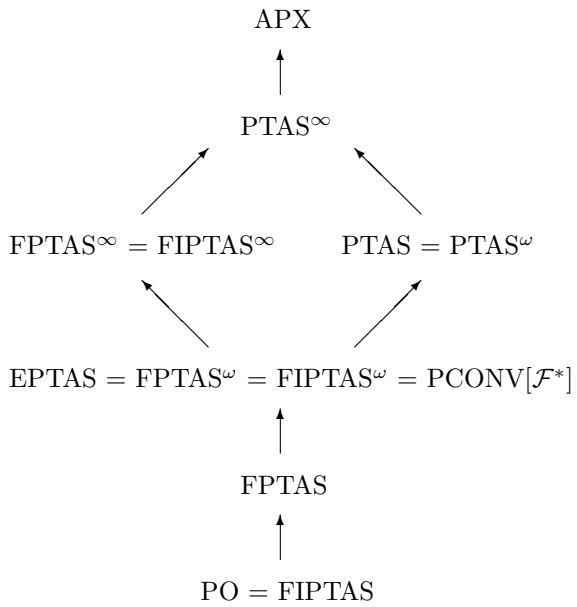

Figure 1: The arrows represent the 'is contained in'-relation. The existence of any inclusion relation not in the above graph or the collapse of one of the arrows implies that either P=NP or FPT=W[1].

eptas is called EPTAS. We show that EPTAS is closely related to the classes of problems admitting 'asymptotic' approximation schemes, where the relative error ϵ is attained only asymptotically, i.e. for instances of large size or with a large optimum. Optimum-asymptotic approximation schemes are well-known, for instance from the study of approximation algorithms for bin packing problems (see e.g. [11, 12, 28]).

Concretely, we prove that all commonly distinguished classes of problems with an asymptotic polynomial-time approximation scheme are a superclass of EPTAS. Two of these classes, FPTAS$^\omega$ and FIPTAS$^\omega$, both corresponding to size-asymptotic schemes, even coincide with the class EPTAS (see Section 3). This settles one of the main questions that motivated this paper: recent research [42, 43, 44] had shown that natural problems having an fptas$^\omega$ exist, but their position in the hierarchy of approximable problems was hitherto unclear.

Moreover, we distinguish the notion of convergent polynomial-time approximation schemes, in which the approximation ratio improves by some function of the instance size as the instance size grows. We show that the corresponding class of optimization problems is equivalent to EPTAS as well (see Section 4). This strengthens the assertion that EPTAS is central in the landscape of problems admitting polynomial-time approximation schemes and deepens the understanding of this class.

We also consider the characteristics of asymptotic approximation schemes. In general, fully polynomial-time approximation schemes have a running time depending (polynomially) on both the instance size and $1/\epsilon$. If the running time depends only on the instance size, a scheme is called a fully input-polynomial-time approximation scheme (fiptas). We show that if a problem admits an asymptotic fptas (a fptas$^\infty$ or a fptas$^\omega$), then this problem admits an asymptotic fiptas of the same kind (a fiptas$^\infty$ or a fiptas$^\omega$ respectively). Hence the corresponding classes coincide, demonstrating an important property of the notions of size- and optimum-asymptotic approximation schemes.

Figure 1 shows the hierarchy of problem classes that follows from this paper. A proper definition of all classes in the figure is given in the next sections.

In the second part of the paper, we discuss several ways to overcome the hardness gap between asymptotic approximation schemes and classic approximation schemes. Section 6 employs ideas from fixed-parameter complexity theory. The results of this section lead to a new characteriza-

tion of problems having a ptas or fptas by means of fixed-parameter tractability and optimum-asymptotic approximation schemes. This characterization is subsequently used to prove that a large number of problems cannot have an optimum-asymptotic polynomial-time approximation scheme (ptas$^\infty$) unless P=NP. This includes all MAX-SNP-complete problems. From Figure 1 we can see that this strengthens a result from Arora et al. [1], who only proved that such problems cannot have a ptas.

Additionally, we study reductions that preserve approximability by optimum-asymptotic approximation schemes. We show that several results on the nonexistence of optimum-asymptotic polynomial-time approximation schemes in the literature implicitly use such a reduction and thus follow from the general approach presented here. Furthermore, we prove that Minimum Bin Packing cannot have such a reduction from Maximum Satisfiability unless P=NP. This augments results by Crescenzi et al. [13], who showed that no approximation-preserving reduction exists in this case unless the polynomial hierarchy collapses.

Finally, we propose the notion of pumpability in Section 7. Problems having asymptotic polynomial-time approximation schemes can sometimes be 'pumped' to a form that admits a ptas, eptas, or even an fptas if the optimization problem under consideration is pumpable. This is useful for completing Figure 1, but also for improving the effectiveness of asymptotic approximation schemes. Furthermore, we provide insight into which problems are pumpable and show for instance that all problems in MAX-SNP are pumpable.

2 Preliminaries

To make formal statements about equivalences among classes of approximation schemes, we have to be precise about the machine model we use, the type of problems that are considered, and the definitions of the studied classes. Throughout the paper, we assume the basic random access machine model with logarithmic costs and representations in bits, which implies that within cost (time) t, the machine can output at most t bits. This machine model is polynomially equivalent to the classic Turing machine and thus defines the classic complexity classes up to polynomial time factors. Furthermore, all numbers are assumed to be rationals, unless otherwise specified.

Using this model, we study optimization problems following the definitions as can be found for instance in Ausiello et al. [2].

Definition 2.1 *An* optimization problem P *is characterized by four properties:*

- *a set of* instances *(bitstrings)* I_P*;*
- *a function* S_P *that maps instances of* P *to (nonempty) sets of* feasible solutions *(bitstrings) for these instances;*
- *an* objective function m_P *that gives for each pair* (x, y) *consisting of instance* $x \in I_P$ *and solution* $y \in S_P(x)$ *a positive integer* $m_P(x, y)$*, the* objective value*;*
- *a* goal $\text{goal}_P \in \{\min, \max\}$ *depending on whether* P *is a minimization or a maximization problem.*

We denote by $S_P^*(x) \subseteq S_P(x)$ the set of optimal solutions for an instance $x \in I_P$, i.e. $S_P^*(x)$ consists of all $y^* \in S_P(x)$ for which

$$m_P(x, y^*) = \text{goal}_P\{m_P(x, y) \mid y \in S_P(x)\}.$$

The objective function value attained by the optimal solutions for an instance x is denoted $m_P^*(x)$.

Definition 2.2 *An optimization problem* P *is in the class* NPO *if*

- *the set of instances* I_P *can be recognized in polynomial time;*

- *there is a (monotone nondecreasing) polynomial q_P such that $|y| \leq q_P(|x|)$ for any instance $x \in I_P$ and any feasible solution $y \in S_P(x)$;*
- *for any instance $x \in I_P$ and any y with $|y| \leq q_P(|x|)$, one can decide in polynomial time whether $y \in S_P(x)$;*
- *there is a (monotone nondecreasing) polynomial r_P such that the objective function m_P is computable in $r_P(|x|, |y|)$ time for any $x \in I_P$ and $y \in S_P(x)$.*

Note that for any problem $P \in$ NPO and any $n \in \mathbb{N}$ the maximum objective value of instances of size n, i.e. $\max\{m_P(x, y) \mid x \in I_P, |x| = n, y \in S_P(x)\}$, is bounded by $2^{r_P(n, q_P(n))}$, as the objective function value of any $x \in I_P$ and $y \in S_P(x)$ can be represented by at most $r_P(|x|, |y|) \leq r_P(|x|, q_P(|x|))$ bits. Let $M_P(n) = 2^{r_P(n, q_P(n))}$.

Lemma 2.3 *For any NPO-problem P, for any $x \in I_P$, and for any $n \in \mathbb{N}$, if $m^*_P(x) > M_P(n)$, then $|x| > n$.*

All problems considered below will be in NPO and all considered classes will be subclasses of NPO. From now on, we drop the subscript P if P is clear from the context.

If one equates NPO to NP, then PO is the equivalent of P. *PO* is the class of problems in NPO for which an optimal solution $y^* \in S^*(x)$ can be computed in time polynomial in $|x|$ for any $x \in I$. Paz and Moran [38] proved that P=NP implies PO=NPO and vice versa. Because it is not expected that all problems in NPO also fall in PO, several classes have been defined that contain NPO-problems for which an *approximate* solution can be found in polynomial time. Approximation algorithms are classified by two properties: their running time and their approximation ratio.

Definition 2.4 ([2, 22]) *For an optimization problem $P \in$ NPO, any $x \in I_P$, and any $y \in S(x)$, the* approximation ratio *achieved by y for x is*

$$R(x, y) = \max\left\{\frac{m(x, y)}{m^*(x)}, \frac{m^*(x)}{m(x, y)}\right\}.$$

We say that y is within (a factor) *r of $m^*(x)$ if $R(x, y) \leq r$. The* approximation ratio *of an algorithm $\mathcal{A}$ is defined as*

$$R_{\mathcal{A}} = \max\{R(x, \mathcal{A}(x)) \mid x \in I_P\}.$$

Any textbook on approximation algorithms covers at least the classes of Table 1. The table should be interpreted as follows: *PTAS*, for instance, is the class of optimization problems P in NPO having a *ptas*, i.e. having an algorithm $\mathcal{A}$ such that for any instance $x \in I_P$ and any $\epsilon > 0$, $\mathcal{A}(x, \epsilon)$ runs in time polynomial in $|x|$ for every fixed ϵ and the solution output by $\mathcal{A}(x, \epsilon)$ has approximation ratio $(1 + \epsilon)$. We use lower-case letters for a scheme name and upper-case letters for the name of the corresponding class (i.e. ptas and PTAS).

The class FIPTAS (Fully Input-Polynomial-Time Approximation Scheme) in Table 1 is a new class. Clearly, FIPTAS=PO (use $\epsilon = 1/M(|x|)$), but the reason for defining this class will become apparent later.

A relatively new class of increasing interest is EPTAS [3, 7].

Definition 2.5 *Algorithm $\mathcal{A}$ is an* efficient polynomial-time approximation scheme (eptas) *for problem $P \in$ NPO if there is a computable function $f : \mathbb{Q}_{\geq 1} \to \mathbb{N}$ such that for any $x \in I_P$ and any $\epsilon > 0$, $\mathcal{A}(x, \epsilon)$ runs in time $f(1/\epsilon)$ times a fixed polynomial in $|x|$ and the solution output by $\mathcal{A}(x, \epsilon)$ has approximation ratio $(1 + \epsilon)$. An NPO-problem is in the class EPTAS if and only if it has an eptas.*

The popularity of eptas is not only due to the separate dependence on $1/\epsilon$ and instance size in the running time, but also to the beautiful relation to the widely researched class FPT: any problem admitting an eptas is also in FPT in its standard parameterization [3, 7]. An interesting exploration of the type of problems that admits an eptas may be found in Cai et al. [5].

Problem class	Running time	Approx. ratio
APX	Polynomial in $\|x\|$	c
PTAS	Polynomial in $\|x\|$ (for every fixed ϵ)	$(1+\epsilon)$
FPTAS	Polynomial in $\|x\|$ and $1/\epsilon$	$(1+\epsilon)$
FIPTAS	Polynomial in $\|x\|$	$(1+\epsilon)$
PO	Polynomial in $\|x\|$	1

Table 1: Problem classes and the distinguishing properties of the approximation algorithms admitted by problems in a particular class.

It is well-known that PO $\subseteq$ FPTAS $\subseteq$ EPTAS $\subseteq$ PTAS $\subseteq$ APX $\subseteq$ NPO. In most cases, the inclusion is strict (unless P=NP), except that EPTAS $\subset$ PTAS unless FPT=W[1] [3, 7]. The question whether FPT=W[1] is an open problem in fixed-parameter complexity theory akin to the question whether P=NP in classic complexity theory (see e.g. Downey and Fellows [17]).

3 Asymptotic Approximation Schemes

Informally, an approximation scheme is *asymptotic* if it gives a $(1+\epsilon)$-approximation under a condition that is asymptotically true. We study two types of asymptotic approximation schemes. We first consider approximation schemes where the size of the instance needs to be large enough. The other type is treated in Section 5.

Definition 3.1 *An approximation scheme $\mathcal{A}$ for $P \in NPO$ is* size-asymptotic *if there is a computable function $a : \mathbb{Q}_{\geq 1} \to \mathbb{N}$ (the* threshold function*) such that for any $\epsilon > 0$ and any $x \in I_P$, it returns a $y \in S(x)$ and if $|x| \geq a(1/\epsilon)$, then y is within $(1+\epsilon)$ of $m^*(x)$.*

This definition leads to the following classes of size-asymptotic approximation schemes.

Problem class	Running time	Approx. ratio
PTAS$^\omega$	Polynomial in $\|x\|$ (for every fixed ϵ)	$(1+\epsilon)$ if $\|x\| \geq a(1/\epsilon)$
FPTAS$^\omega$	Polynomial in $\|x\|$ and $1/\epsilon$	$(1+\epsilon)$ if $\|x\| \geq a(1/\epsilon)$
FIPTAS$^\omega$	Polynomial in $\|x\|$	$(1+\epsilon)$ if $\|x\| \geq a(1/\epsilon)$

Example 3.2 *Maximum Independent Set has a fiptas$^\omega$ on bounded-ply disk graphs [43, 44]. Disk graphs are intersection graphs of disks in the plane, i.e. given a set of disks, each vertex of the graph corresponds to a disk and there is an edge between two vertices if the corresponding disks intersect. A set of disks has ply γ if γ is the smallest integer such that any point of the plane is overlapped by at most γ disks. One can find in $O(|x|^{10}\log^4|x|)$ time an independent set of an instance x of this problem. If an odd integer k can be chosen such that $\max\{5, 4(1+\epsilon)/\epsilon\} \leq k \leq c_1 \log|x|/\log(c_2\gamma)$ (where c_1, c_2 are fixed constants), then this independent set will be within $(1+\epsilon)$ of the optimum. If $\gamma = \gamma(|x|) = O(|x|^{o(1)})$, such an integer exists if $|x| \geq a(1/\epsilon)$ for some function a.*

We start with some easy observations about the size-asymptotic classes.

Proposition 3.3 *The following relations hold:*

- *FIPTAS$^\omega$ $\subseteq$ FPTAS$^\omega$ $\subseteq$ PTAS$^\omega$ and*
- *FIPTAS $\subseteq$ FIPTAS$^\omega$, FPTAS $\subseteq$ FPTAS$^\omega$, PTAS $\subseteq$ PTAS$^\omega$.*

The relations given by this proposition are straightforward and one might expect that the inclusions are strict under some hardness condition. However, this turns out not to be true for all of them. We can prove some very interesting equivalences and tie these new classes to existing approximation classes, in particular to EPTAS.

Theorem 3.4 *EPTAS = FPTAS$^\omega$ = FIPTAS$^\omega$.*

Proof: We first show that EPTAS $\subseteq$ FIPTAS$^\omega$. Let $P \in$ EPTAS and let $\mathcal{A}$ be an eptas for P with running time at most $p(|x|) \cdot f(1/\epsilon)$ for some computable function f and polynomial p. Construct a fiptas$^\omega$ for P as follows. Given an arbitrary instance $x \in I_P$ and an arbitrary $\epsilon > 0$, run $\mathcal{A}(x, \epsilon)$ for $p(|x|) \cdot |x|$ time steps. If $\mathcal{A}(x, \epsilon)$ finishes, return the solution given by $\mathcal{A}(x, \epsilon)$. Otherwise, return $\mathcal{A}(x, 1/2)$. This algorithm clearly runs in time polynomial in $|x|$ and always returns a feasible solution. Furthermore if $|x| \geq f(1/\epsilon)$, $\mathcal{A}(x, \epsilon)$ always finishes and returns a feasible solution with approximation ratio $(1 + \epsilon)$. Hence we constructed a fiptas$^\omega$ for P with $a = f$.

We next prove that FPTAS$^\omega$ $\subseteq$ EPTAS. Let $P \in$ FPTAS$^\omega$ and let $\mathcal{A}$ be an fptas$^\omega$ for P with threshold function a. Construct an eptas as follows. Given an arbitrary instance $x \in I_P$ and an arbitrary $\epsilon > 0$, compute $a(1/\epsilon)$. By assumption, $a(1/\epsilon)$ is computable. The amount of time it takes to compute $a(1/\epsilon)$ is some computable function depending on $1/\epsilon$. If $|x| \geq a(1/\epsilon)$, simply compute and return $\mathcal{A}(x, \epsilon)$ in time polynomial in $|x|$ and $1/\epsilon$. If $|x| < a(1/\epsilon)$, proceed as follows. As FPTAS$^\omega$ $\subseteq$ NPO, any feasible solution for x has size at most $q(|x|)$ for some fixed polynomial q. Furthermore, given any y with $|y| \leq q(|x|)$, one can determine in polynomial time whether $y \in S_P(x)$. The objective value of a feasible solution can also be computed in polynomial time. Hence by employing exhaustive search, one can find a $y^* \in S_P^*(x)$ in time $\text{poly}(|x|) \cdot 2^{q(|x|)} \cdot r_P(|x|, q(|x|)) = 2^{q(a(1/\epsilon))} \cdot \text{poly}(a(1/\epsilon))$. The result is an eptas for P with appropriately defined function f.

Since FIPTAS$^\omega$ $\subseteq$ FPTAS$^\omega$, we have that EPTAS $\subseteq$ FIPTAS$^\omega$ $\subseteq$ FPTAS$^\omega$ $\subseteq$ EPTAS, and hence the classes must be equal. $\square$

The exponential increase in running time in the reduction from an fptas$^\omega$ to an eptas might be reduced by using an exact or fixed-parameter algorithm specific to the problem. As we show in Section 7, one can avoid such an increase altogether for many problems.

The equivalence of F(I)PTAS$^\omega$ and EPTAS allows an indirect proof of the existence of an eptas for a problem, where a direct proof seems more difficult.

Example 3.5 *Maximum Independent Set on disk graphs of bounded ply has a fiptas$^\omega$ (Example 3.2) and thus, as a consequence of Theorem 3.4, an eptas.*

We now show that PTAS$^\omega$ and PTAS are in fact equivalent as well.

Theorem 3.6 *PTAS = PTAS$^\omega$.*

Proof: By Proposition 3.3 it suffices to prove that PTAS$^\omega$ $\subseteq$ PTAS. Let $P \in$ PTAS$^\omega$ and let $\mathcal{A}$ be a ptas$^\omega$ for P. For an arbitrary instance $x \in I_P$ and an arbitrary $\epsilon > 0$, compute $a(1/\epsilon)$. If $|x| \geq a(1/\epsilon)$, compute and return $\mathcal{A}(x, \epsilon)$. Otherwise, apply the same exhaustive search technique as in the proof of Theorem 3.4. The result is a ptas for P. $\square$

4 Convergent Approximation Schemes

Size-asymptotic approximation schemes have a threshold function, depending on $1/\epsilon$, such that a good approximate solution is guaranteed if the size of the instance is larger than the threshold. It seems then that the quality of the computed solution can be arbitrarily bad for small instances, while from a certain instance size onward, the quality suddenly becomes very good. Practical examples of size-asymptotic approximation schemes show however that the approximation ratio can improve steadily as the instance size increases and eventually converges (to 1).

Surprisingly, this also holds in general. In this section, we define and study these convergent approximation schemes more precisely. The main result is that a problem has a fptas$^\omega$ if and only if it also has a convergent approximation scheme.

In the following, we use $\mathcal{F}^*$ to denote the family of all monotone nondecreasing computable functions $f : \mathbb{N} \to \mathbb{Q}_{\geq 1}$ with $\liminf_{n\to\infty} f(n) = \infty$. Let $\mathcal{P}$ denote the family of those functions in $\mathcal{F}^*$ that are bounded by a (monotone) polynomial.

Definition 4.1 *Let $f \in \mathcal{F}^*$. An approximation scheme $\mathcal{A}$ for $P \in NPO$ is said to be* ϵ-convergent *w.r.t. f if for any $\epsilon > 0$ and any $x \in I_P$, $\mathcal{A}(x, \epsilon)$ returns a $y \in S(x)$ within $(1+\epsilon/f(|x|))$ of $m^*(x)$.*

This definition gives rise to schemes ptas[f], fptas[f], and fiptas[f] and the classes PTAS[f], FPTAS[f], and FIPTAS[f] in the natural way. We also define a special subclass for the case when $\epsilon = 1$.

Definition 4.2 *Let $f \in \mathcal{F}^*$. Algorithm $\mathcal{A}$ is a* convergent polynomial-time approximation scheme w.r.t. f (denoted as pconv[f]) *if for any $x \in I$, $\mathcal{A}(x)$ runs in time polynomial in $|x|$ and returns a $y \in S(x)$ within $(1 + 1/f(|x|))$ of $m^*(x)$. The corresponding class is* PCONV[f].

Example 4.3 *Chiba et al. [10] give a pconv[$O(\sqrt{\log\log|x|})$] for Maximum Independent Set on planar graphs. This follows from a general $O(|x|\log|x|)$ algorithm giving a $1 + 1/O(\sqrt{\log\log|x|})$-approximation for several hereditary problems on planar graphs. Demaine et al. [16] give a pconv[$\log|x|$][1] for Maximum Independent Set on single-crossing-minor-free graphs (a generalization of planar graphs).*

Definition 4.4 *For any family of functions $\mathcal{F} \subseteq \mathcal{F}^*$, let PCONV[$\mathcal{F}$] $= \bigcup_{f\in\mathcal{F}}$ PCONV[f]. We similarly define PTAS[$\mathcal{F}$], FPTAS[$\mathcal{F}$], and FIPTAS[$\mathcal{F}$].*

We first state some straightforward relations.

Proposition 4.5 *The following relations hold:*

- *FIPTAS[$\mathcal{F}^*$] = PO, FPTAS[$\mathcal{F}^*$] $\subseteq$ FPTAS, PTAS[$\mathcal{F}^*$] $\subseteq$ PTAS,*
- *for any $f \in \mathcal{F}^*$, FIPTAS[f] $\subseteq$ FPTAS[f] $\subseteq$ PTAS[f] $\subseteq$ PCONV[f], and*
- *for any $f, f' \in \mathcal{F}^*$ with $f(n) \le f'(n)$ for any $n \in \mathbb{N}$, FIPTAS[f'] $\subseteq$ FIPTAS[f], FPTAS[f'] $\subseteq$ FPTAS[f], PTAS[f'] $\subseteq$ PTAS[f], and PCONV[f'] $\subseteq$ PCONV[f].*

Looking closely at the papers cited in Example 4.3, one can observe that the algorithms they describe are actually a pconv[f] (for certain f) as well as an eptas. This is not a coincidence!

Lemma 4.6 *If $P \in$ FPTAS$^\omega$, then $P \in$ PCONV[f] for some $f \in \mathcal{P}$.*

Proof: Let $P \in$ FPTAS$^\omega$ and let $\mathcal{A}$ be an fptas$^\omega$ for P delivering a $(1 + \epsilon)$-approximate solution if $|x| \ge a(1/\epsilon)$ for some computable function a. We use $\mathcal{A}$ to construct a pconv[f] for a suitably chosen function f.

Fix a monotone, polynomially-bounded function $p(n)$ (for instance $p(n) = n$). Because a is computable, we can compute

$$c_1 := 0, c_2 := a(2), c_3 := a(3), c_4 := a(4), \ldots$$

Compute the values of this series (starting at c_1, then $c_2, c_3, \ldots$) for at most $p(n)$ time steps total. Let $f(n)$ be the highest index k that satisfies $n \ge c_k$ among the fully computed values c_k. Because $c_1 = 0$, this function is properly defined. Furthermore, f is a monotone, computable function with $\liminf_{n\to\infty} f(n) = \infty$ and $f(n) \le p(n)$.

Observe that because FPTAS$^\omega \subseteq$ PTAS, P must have a 2-approximation algorithm $\mathcal{B}$ running in time polynomial in the size of the input. Now consider the following algorithm $\mathcal{A}'(x)$ for instances $x \in I_P$: if $f(|x|) = 1$, return $\mathcal{B}(x)$, otherwise return $\mathcal{A}(x, \epsilon)$ with $\epsilon = 1/f(|x|)$. We claim that $\mathcal{A}'(x)$ is a pconv[f] for P with f as defined above.

Let y be the solution output by $\mathcal{A}'(x)$ for some $x \in I_P$. Clearly, $y \in S(x)$. If $f(|x|) = 1$, $\mathcal{A}'(x)$ trivially returns a $(1 + 1/f(|x|))$-approximation. If $f(|x|) > 1$, then because $|x| \ge c_{f(|x|)} = a(f(|x|))$, y is a $(1 + 1/f(|x|))$-approximate solution. Furthermore, the running time is

$$O(\text{poly}(1/\epsilon) \cdot \text{poly}(|x|)) = O(\text{poly}(f(|x|)) \cdot \text{poly}(|x|)) = O(\text{poly}(p(|x|)) \cdot \text{poly}(|x|)) = O(\text{poly}(|x|)).$$

Note that $f(|x|)$ is also computable in polynomial time. Hence we achieve a $(1 + 1/f(|x|))$-approximation in time polynomial in $|x|$ and we thus have that $P \in$ PCONV[f]. □

[1] Formally, this should be $\lfloor \log|x| \rfloor$. Throughout the paper we ignore these technicalities to improve legibility.

The lemma implies that FPTAS$^\omega \subseteq$ PCONV[$\mathcal{F}^*$]. By Theorem 3.4, this in turn implies that EPTAS $\subseteq$ PCONV[$\mathcal{F}^*$]. We can give a direct proof of this consequence, which has the additional advantage that the function of $1/\epsilon$ in the running time of the eptas is only needed for the analysis and not for the algorithmic part of the reduction.

Lemma 4.7 *If $P \in$ EPTAS, then $P \in$ PCONV[f] for some $f \in \mathcal{P}$.*

Proof: Let $\mathcal{A}$ be an eptas for P, which runs in time $g(1/\epsilon) \cdot p(|x|)$ for any $x \in I_P$ and any $\epsilon > 0$ for some function g and polynomial p. We assume that g and p are computable and that p is known (g is not necessarily known). Now run $y_1 = \mathcal{A}(x, 1)$ to completion and $y_2 = \mathcal{A}(x, 1/2), \ldots,$ $y_{|x|+1} = \mathcal{A}(x, 1/(|x| + 1))$ for at most $|x| \cdot p(|x|)$ time steps each, and return $\text{goal}_P\{y_k\}$ over all k for which $\mathcal{A}$ finished within the alotted time. This algorithm clearly always outputs a feasible solution in polynomial time. We claim that it yields a $(1 + 1/f(|x|))$-approximation for a suitably chosen monotone computable function f with $\liminf_{n\to\infty} f(n) = \infty$.

We construct the function f as a piecewise constant function on a sequence of intervals $[n_1, n_2)$, $[n_2, n_3), \ldots$. Define $n_1 = 0$ and for any $i \geq 2$,

$$n_i = \max\{n_{i-1} + 1, g(i)\}.$$

Define

$$f(n) = i \quad \text{if } n \in [n_i, n_{i+1}) \text{ for some } i \geq 1.$$

Clearly, $n_i \geq n_{i-1} + 1$ and n_i is finite for any i. Hence f is a monotone nondecreasing function with $\liminf_{n\to\infty} f(n) = \infty$ and $f(n) \leq n + 1$ for any $n \in \mathbb{N}$. Hence $f \in \mathcal{P}$. Furthermore, $f(n)$ is computable, as we only need to compute n_i for a finite number (at most n) values of i and g is computable.

If $f(|x|) = 1$ then, since $\mathcal{A}(x, 1)$ runs to completion, the algorithm delivers at least a $(1 + 1/f(|x|))$-approximation. If $f(|x|)$ is equal to i for some $i \geq 2$, then by the construction of f and n_i, $|x| \geq n_i \geq g(i)$ and $i \leq |x| + 1$. Hence certainly $\mathcal{A}(x, 1/i)$ runs to completion within the time limit set for it and the algorithm returns at least a $(1 + 1/i)$-approximation of the optimum. But then the algorithm returns at least a $(1 + 1/f(|x|))$-approximation of the optimum. □

We now prove the converse relation.

Lemma 4.8 *If $P \in$ PCONV[$\mathcal{F}^*$], then $P \in$ FIPTAS$^\omega$.*

Proof: Let $P \in$ PCONV[f] and let $\mathcal{A}$ be a pconv[f] for P with $f \in \mathcal{F}^*$. We claim that $\mathcal{A}$ is a fiptas$^\omega$ for P with threshold function

$$a(1/\epsilon) := \begin{cases} 0 & \text{if } 1/\epsilon \leq f(0) \\ \max\{n \in \mathbb{N} \mid 1/\epsilon > f(n)\} & \text{otherwise.} \end{cases}$$

Note that a is indeed properly defined, as f is monotone and $\liminf_{n\to\infty} f(n) = \infty$.

Clearly, a is a computable function and algorithm $\mathcal{A}(x)$ runs in time polynomial in $|x|$ for any $x \in I_P$ and any $\epsilon > 0$. Furthermore, if $|x| \geq a(1/\epsilon)$, then $\epsilon \geq 1/f(|x|)$ by the definition of a. Hence $(1 + 1/f(|x|)) \leq (1 + \epsilon)$ and thus $\mathcal{A}(x)$ returns a $(1 + \epsilon)$-approximate solution. □

Note that the function a in the proof of Lemma 4.8 is essentially the inverse of f (it is not precisely the inverse, as f might not be invertible). Similarly, the function f constructed in the proof of Lemma 4.7 is the 'inverse' of g.

Using Theorem 3.4 and Lemmas 4.7 and 4.8, we obtain the following key theorem.

Theorem 4.9 *EPTAS = FPTAS$^\omega$ = FIPTAS$^\omega$ = PCONV[$\mathcal{F}^*$].*

To state this theorem informally: for polynomial-time approximation schemes, having a single factor depending (only) on $1/\epsilon$ in the running time is equivalent to having such a function as a threshold for yielding an $(1 + \epsilon)$-approximation, which in turn is equivalent to having the attained approximation ratio improving to 1 as the instance size increases.

4.1 Detailed Relations for Specific Functions

If we have specific knowledge of the function f that appears in the approximation ratio of a convergent approximation scheme, we can prove some more detailed relations. First we consider the family of classes where this function is bounded by a polynomial.

Theorem 4.10 *FPTAS = FPTAS[f] for any $f \in \mathcal{P}$.*

Proof: By Proposition 4.5, it suffices to show that FPTAS $\subseteq$ FPTAS[f] for any $f \in \mathcal{P}$. Let $P \in$ FPTAS and let $\mathcal{A}$ be an fptas for P. Let f be upper bounded by the (monotone) polynomial p, i.e. $f(n) \leq p(n)$ for all $n > 0$. Consider an $x \in I_P$ and an $\epsilon > 0$. Compute $p(|x|)$ and run $\mathcal{A}(x, \epsilon/p(|x|))$. This algorithm runs in time polynomial in $|x|$ and $1/\epsilon$ and returns a $(1+\epsilon/p(|x|)) \leq (1+\epsilon/f(|x|))$-approximate solution. Hence $P \in$ FPTAS[f]. □

Corollary 4.11 $FPTAS[\mathcal{P}] = \bigcup_{f\in\mathcal{P}} FPTAS[f] = \bigcap_{f\in\mathcal{P}} FPTAS[f]$.

A problem $P \in$ NPO is *polynomially-bounded* if there is a polynomial $p : \mathbb{N} \to \mathbb{N}$ such that $m(x, y) \leq p(|x|)$ for all $x \in I_P$ and $y \in S_P(x)$ [21, 2].

Lemma 4.12 *Let P be an NP-hard, polynomially-bounded optimization problem and p the corresponding bounding (monotone) polynomial plus 1. Then $P \notin$ PCONV[p], unless P=NP.*

Proof: Assume that P is a minimization problem (the case when P is a maximization problem is similar). Suppose that $P \in$ PCONV[p]. Then there exists an algorithm $\mathcal{A}$ such that for any instance x of P, $\mathcal{A}(x)$ runs in time polynomial in $|x|$ and delivers a feasible solution y with approximation ratio $1 + 1/p(|x|)$. But then

$$m(x, y) \leq (1 + 1/p(|x|)) \cdot m^*(x) < m^*(x) + 1.$$

But this is only possible if $m(x, y) = m^*(x)$. This means that we found an optimal solution in polynomial time. This is impossible unless P=NP. □

Using this lemma and the easy fact that FPTAS $\subseteq$ PCONV[f] for any $f \in \mathcal{P}$, we can show the following corollary, which is well-known and easy [21].

Corollary 4.13 *No NP-hard, polynomially-bounded optimization problem admits an fptas, unless P=NP.*

Proof: Let P be an NP-hard, polynomially-bounded optimization problem and p the corresponding bounding polynomial plus 1. Then $P \notin$ PCONV[p] and thus $P \notin$ FPTAS, unless P=NP. □

For functions in the approximation ratio of convergent schemes that are not polynomial, we can also prove some interesting relations. As noted before, the function f constructed in the proof of Lemma 4.7 is the 'inverse' of the function g in the running time of the eptas. This leads to the following general result.

Theorem 4.14 *If g is an invertible function, then a problem that has an eptas with running time $g(1/\epsilon) \cdot |x|^{O(1)}$ also has a pconv[$g^{-1}(|x|)$].*

For instance, if $g(1/\epsilon) = 2^{2^{1/\epsilon}}$, then the problem has a pconv[$\log\log|x|$]. The statement of this theorem is not 'if and only if', because Lemma 4.8 only gives a fiptas$^\omega$. Transforming it to an eptas using Theorem 3.4 increases the running time exponentially in g.

For some functions, one can even improve on Theorem 4.14.

Theorem 4.15 *For any polynomial p of degree s, a problem that has an eptas with running time bounded by $2^{p(1/\epsilon)} \cdot |x|^{O(1)}$ also has a ptas[$\log^{1/s}|x|$].*

It seems unlikely that an equivalence as in Theorem 4.15 will also hold if g is doubly-exponential.

5 Optimum-Asymptotic Approximation Schemes

Approximation schemes that give a $(1+\epsilon)$-approximation if the optimum is large enough are quite common. They are particularly well-known for Minimum Bin Packing (see e.g. [11, 12, 28]), but similar schemes exist for other problems, e.g. for Minimum Degree Spanning Tree [19] and Chromatic Index [45]. There are also several ways to define what constitutes an optimum-asymptotic approximation scheme [2, 12, 22, 27]. We prove that these definitions are actually equivalent. More interestingly, we revisit the relation of optimum-asymptotic approximation schemes to nonasymptotic approximation schemes and show that EPTAS is a subclass of the optimum-asymptotic classes. Finally, we investigate the relations between various types of optimum-asymptotic approximation schemes.

We first define optimum-asymptotic approximation schemes. The definition we use is both in line with previous definitions of optimum-asymptotic approximation schemes (see e.g. [22]) and with the definition of size-asymptotic approximation schemes (see Definition 3.1).

Definition 5.1 *An approximation scheme $\mathcal{A}$ for $P \in NPO$ is* optimum-asymptotic *if there is a computable function $b : \mathbb{Q}_{\geq 1} \to \mathbb{N}$ (the* threshold function*) and an associated constant ϵ_b with the property that $b(1/\epsilon) \leq 1$ for each $\epsilon \geq \epsilon_b$, such that for any $\epsilon > 0$ and any $x \in I_P$, it returns a $y \in S(x)$ and if $m^*(x) \geq b(1/\epsilon)$, then y is within $(1+\epsilon)$ of $m^*(x)$.*

This leads to the definition of ptas$^\infty$, fptas$^\infty$, and fiptas$^\infty$ schemes and to classes PTAS$^\infty$, FPTAS$^\infty$, and FIPTAS$^\infty$, all defined as expected.

Example 5.2 *Karmarkar and Karp [28] give a fiptas$^\infty$ for Minimum Bin Packing. For an instance x, the returned solution has objective value at most $m^*(x) + \log^2 m^*(x)$ and is found in $\widetilde{O}(|x|^8)$ time, where the $\widetilde{O}$ hides certain polylogarithmic terms.*

Note that optimum-asymptotic schemes are defined analogously to size-asymptotic schemes, except for the extra requirement on $b(1/\epsilon)$. This technicality seems to be indispensable when trying to prove that optimum-asymptotic approximation scheme classes are a subclass of APX and behave as the known asymptotic approximation classes. In particular, it facilitates the following crucial property.

Lemma 5.3 *Let $\mathcal{A}$ be a ptas$^\infty$ for a problem P and let ϵ_b be the constant associated with $\mathcal{A}$'s threshold function b. Then P has a polynomial-time $(1+\epsilon_b)$-approximation algorithm.*

This follows immediately from the fact that $m^*(x) \geq 1 \geq b(1/\epsilon_b)$ for any $x \in I_P$.

Corollary 5.4 *PTAS$^\infty$ $\subseteq$ APX.*

It follows from Queyranne [40] that this inclusion is strict unless P=NP (see also Theorem 7.7).

5.1 Equivalence of Definitions

Lemma 5.3 can be used to prove that the definition of optimum-asymptotic approximation schemes yields classes that are polynomially equivalent to the asymptotic approximation classes defined in the literature [2, 12] .

Definition 5.5 *Algorithm $\mathcal{A}$ is an* asymptotic polynomial-time approximation scheme (aptas) *for P if for any $\epsilon > 0$ there is a computable constant c_ϵ such that for any $x \in I_P$, $\mathcal{A}(x, \epsilon)$ runs in time polynomial in $|x|$ for every fixed ϵ and the solution y output by $\mathcal{A}(x, \epsilon)$ is feasible and within $(1+\epsilon) + c_\epsilon/m^*(x)$ of $m^*(x)$. A problem is in the class* APTAS *if and only if it has an aptas.*

One similarly defines afptas and afiptas and corresponding classes AFPTAS and AFIPTAS in the natural way. These classes are sometimes also referred to as PTAAS, FPTAAS, and FIPTAAS, for (Fully-)(Input-)Polynomial-Time Asymptotic Approximation Scheme [12]. The class APTAS has also been called ASY-PTAS [46].

Example 5.6 *Coffman and Lueker [12] present an AFPTAS (or FPTAAS) for Extensible Bin Packing with* $c_\epsilon = O(1/\epsilon \log 1/\epsilon)$.

Theorem 5.7 *APTAS=PTAS*$^\infty$*, AFPTAS=FPTAS*$^\infty$*, and AFIPTAS=FIPTAS*$^\infty$.

Proof: Consider a problem $P \in$ APTAS and an aptas $\mathcal{A}$ for P. It is well-known (and easily proved) that APTAS $\subseteq$ APX [2], so let $\mathcal{B}$ be a polynomial-time c-approximation algorithm for P for some constant $c > 1$. We claim that for any $x \in I_P$ and any $\epsilon > 0$, the solution attaining $\text{goal}_P\{m(x, \mathcal{A}(x, \epsilon/2)), m(x, \mathcal{B}(x))\}$ is a ptas$^\infty$ with threshold function

$$b(1/\epsilon) = \begin{cases} 1 & \text{if } \epsilon \geq c - 1 \\ c_{\epsilon/2} \cdot 2/\epsilon & \text{otherwise.} \end{cases}$$

and associated constant $\epsilon_b := c - 1$.

The function b is obviously computable, since c_ϵ is computable. As $\mathcal{A}$ and $\mathcal{B}$ run in time polynomial in $|x|$ for any instance $x \in I_P$ and every fixed $\epsilon > 0$, it remains to show that the returned solution is a $(1 + \epsilon)$-approximate solution on instances $x \in I_P$ if $m^*(x) \geq b(1/\epsilon)$. If $\epsilon \geq \epsilon_b$, then $\mathcal{B}(x)$ ensures a feasible solution within c and thus within $(1+\epsilon)$ of $m^*(x)$. For $\epsilon < \epsilon_b$, a feasible solution is returned and if $m^*(x) \geq c_{\epsilon/2} \cdot 2/\epsilon$, then $(1 + \epsilon/2) + c_{\epsilon/2}/m^*(x) \leq (1 + \epsilon)$, assuring that $\mathcal{A}(x, \epsilon/2)$ delivers a $(1 + \epsilon)$-approximation. This implies that $P \in$ PTAS$^\infty$.

Next we consider a problem $P \in$ PTAS$^\infty$ and a ptas$^\infty$ $\mathcal{A}$ for P with threshold function b and associated constant ϵ_b. By Lemma 5.3, P also has a polynomial-time $(1 + \epsilon_b)$-approximation algorithm $\mathcal{B}$. For any instance $x \in I_P$ and any $\epsilon > 0$, we claim that the algorithm returning the solution attaining $\text{goal}_P\{m(x, \mathcal{A}(x, \epsilon)), m(x, \mathcal{B}(x))\}$ is an aptas with $c_\epsilon = \epsilon_b \cdot b(1/\epsilon)$. If $m^*(x) \geq b(1/\epsilon)$, then $\mathcal{A}(x, \epsilon)$ guarantees a $(1 + \epsilon)$-approximate solution. If $m^*(x) \leq b(1/\epsilon)$, then $\mathcal{B}(x)$ guarantees a $(1 + \epsilon_b)$-approximate solution. Note that

$$1 + \epsilon_b \leq (1 + \epsilon) + \epsilon_b \cdot b(1/\epsilon)/m^*(x) = (1 + \epsilon) + c_\epsilon/m^*(x).$$

Hence $P \in APTAS$.

Similar proofs can be used to show that AFPTAS=FPTAS$^\infty$ and AFIPTAS=FIPTAS$^\infty$. □

Because of these equivalences, all complexity results proved below for PTAS$^\infty$, FPTAS$^\infty$, and FIPTAS$^\infty$ also hold for the classes APTAS, AFPTAS, and AFIPTAS respectively.

We can also make an interesting observation about the class of problems that can be approximated within a constant absolute error.

Definition 5.8 *A problem P can be approximated within a* constant absolute error *if there exists an algorithm $\mathcal{A}$ and constant $c \geq 0$ such that for any $x \in I_P$, $\mathcal{A}(x)$ runs in time polynomial in $|x|$ and the solution y output by $\mathcal{A}$ is feasible satisfies $|m(x, y) - m^*(x)| \leq c$.*

Example 5.9 *Fürer and Raghavachari [19] give a polynomial-time algorithm that approximates Minimum Degree Spanning Tree within constant absolute error* 1.

We now prove that all problems admitting an algorithm with constant absolute error must have a fiptas$^\infty$ with a threshold function that is bounded by a linear function, and vice versa.

Theorem 5.10 *A problem P can be approximated in polynomial time within a constant absolute error if and only if it has a fiptas$^\infty$ with a threshold function b that is bounded by a linear function.*

Proof: Suppose that P can be approximated within a constant absolute error $c \geq 0$. Hence it has a $(c+1)$-approximation algorithm. It then follows from the proof of Theorem 5.7 that P has a fiptas$^\infty$ with $b(1/\epsilon) = 2c/\epsilon$ and associated constant $\epsilon_b = c$.

For the converse, let $\mathcal{A}$ be a fiptas$^\infty$ for P with a threshold function b that is bounded by a linear function. Without loss of generality, we may assume that

$$b(1/\epsilon) = \begin{cases} 1 & \text{if } \epsilon \geq \epsilon_b \\ c/\epsilon & \text{otherwise.} \end{cases}$$

for constants $\epsilon_b \geq 0$ and $c \geq 1$. Then we can also assume that $\epsilon_b = c - 1$ by adjusting ϵ_b or c. By Lemma 5.3, P has a polynomial-time $(1 + \epsilon_b)$-approximation algorithm $\mathcal{B}$. For any instance $x \in I_P$, compute $y' = \mathcal{B}(x)$ in polynomial time and let $s := m(x, y')$.

Consider the case in which $\text{goal}_P = \min$. Then $m^*(x) \leq s \leq c \cdot m^*(x)$. Choose $\epsilon = c^2/s$ and compute $y = \mathcal{A}(x, \epsilon)$ in polynomial time. Note that

$$m^*(x) \geq \frac{s}{c} = \frac{c}{c^2/s}$$

and thus $m^*(x) \geq b(1/\epsilon)$. Hence y is within a factor $(1 + \epsilon)$ of $m^*(x)$. But then

$$m(x, y) \leq (1 + c^2/s) \cdot m^*(x) = m^*(x) + c^2 \cdot \frac{m^*(x)}{s} \leq m^*(x) + c^2.$$

We thus have an algorithm that approximates P within a constant absolute error of c^2.

For the case in which $\text{goal}_P = \max$ the proof is similar. Then $s \leq m^*(x) \leq c \cdot s$. Choose $\epsilon = c/s$ and compute $y = \mathcal{A}(x, \epsilon)$ in polynomial time. Note that

$$m^*(x) \geq s = \frac{c}{c/s}$$

and thus $m^*(x) \geq b(1/\epsilon)$. Hence y is within a factor $(1 + \epsilon)$ of $m^*(x)$. But then

$$m^*(x) \leq (1 + c/s) \cdot m(x, y) = m(x, y) + c \cdot \frac{m(x, y)}{s} \leq m(x, y) + c \cdot \frac{m(x, y)}{m^*(x)/c} \leq m(x, y) + c^2.$$

This gives an algorithm that approximates P again within a constant absolute error of c^2. □

5.2 Equivalence and Containment of Optimum-Asymptotic Classes

Consider now the following natural relations.

Proposition 5.11 *The following relations hold:*

- *$FIPTAS^\infty \subseteq FPTAS^\infty \subseteq PTAS^\infty$ and*
- *$FIPTAS \subseteq FIPTAS^\infty$, $FPTAS \subseteq FPTAS^\infty$, $PTAS \subseteq PTAS^\infty$.*

One might hope or expect that for optimum-asymptotic approximation classes analogous relations hold as for size-asymptotic classes. We show that this is only partially true. First, we investigate the relation between PTAS^∞ and PTAS. We know from Theorem 3.6 that $\text{PTAS}^\omega = \text{PTAS}$, but for optimum-asymptotic problems the equivalent result does not hold (unless P=NP). Actually, we prove a stronger result.

Theorem 5.12 *$FIPTAS^\infty \not\subseteq PTAS$, unless P=NP.*

Proof: The minimum degree spanning tree problem admits a fiptas^∞ (see Example 5.9), but cannot have a ptas unless P=NP [22]. Hence $\text{FIPTAS}^\infty \not\subseteq \text{PTAS}$. □

As described in Section 6 and 7 however, for many problems the existence of a (f(i))ptas^∞ *does* imply the existence of a ptas (or better).

Interestingly, there is a close relation between FPTAS^∞ and FIPTAS^∞. In fact, we can prove that the classes are equal.

Theorem 5.13 *$FPTAS^\infty = FIPTAS^\infty$.*

Proof: By Proposition 5.11, it suffices to show that FPTAS$^\infty \subseteq$ FIPTAS$^\infty$. Let $P \in$ FPTAS$^\infty$ and let $\mathcal{A}$ be an fptas$^\infty$ for P, such that for some computable function b and for any $\epsilon > 0$ and any $x \in I_P$, $\mathcal{A}(x, \epsilon)$ runs in at most $\gamma \cdot (1/\epsilon)^s \cdot |x|^t$ time (for constants $\gamma, s, t > 0$) and yields a $(1+\epsilon)$-approximate solution if $m^*(x) \geq b(1/\epsilon)$. Because FPTAS$^\infty \subseteq$ APX, P has a polynomial-time c-approximation algorithm $\mathcal{B}$ for some constant $c > 1$.

Consider an arbitrary instance $x \in I_P$ and let $\epsilon > 0$ be given. Run $\mathcal{A}(x, \epsilon)$ for at most $\gamma \cdot |x|^{2t}$ time steps. If it finished and thus returned a solution y, return the solution attaining $\text{goal}_P\{m(x, y), m(x, \mathcal{B}(x))\}$. Otherwise, just return $\mathcal{B}(x)$. We claim that this algorithm combined with threshold function b' defined as

$$b'(1/\epsilon) = \begin{cases} 1 & \text{if } \epsilon \geq c - 1; \\ \max\{b(1/\epsilon), 1 + M_P((1/\epsilon)^{s/t})\} & \text{otherwise,} \end{cases}$$

and associated constant $\epsilon_{b'} := c - 1$ is a fiptas$^\infty$ for P.

Clearly, b' is a computable function, the running time of the new algorithm is bounded by a polynomial in $|x|$, and the algorithm returns a feasible solution. Assume that $m^*(x) \geq b'(1/\epsilon)$. If $\epsilon \geq \epsilon_b$, then $\mathcal{B}(x)$ returns a feasible solution within c and thus within $(1+\epsilon)$ of $m^*(x)$. Suppose that $\epsilon < \epsilon_b$. Since $m^*(x) \geq b'(1/\epsilon) \geq b(1/\epsilon)$, $\mathcal{A}(x, \epsilon)$ delivers a $(1+\epsilon)$-approximation *if* it runs to completion. So it remains to show that this is indeed the case, i.e. that $\gamma \cdot (1/\epsilon)^s \cdot |x|^t \leq \gamma \cdot |x|^{2t}$. But this follows from the fact that $m^*(x) \geq b'(1/\epsilon) > M_P((1/\epsilon)^{s/t})$ and thus $|x| \geq (1/\epsilon)^{s/t}$ by Lemma 2.3, or $(1/\epsilon)^s \leq |x|^t$. $\square$

A similar idea can be used to tie EPTAS to the optimum-asymptotic approximation classes.

Theorem 5.14 *FIPTAS$^\omega \subseteq$ FIPTAS$^\infty$.*

Proof: Let $P \in$ FIPTAS$^\omega$ and let $\mathcal{A}$ be a fiptas$^\omega$ for P, such that $\mathcal{A}$ delivers a $(1+\epsilon)$-approximation on instances $x \in I_P$ if $|x| \geq a(1/\epsilon)$. As FIPTAS$^\omega \subseteq$ APX, P also has a polynomial-time c-approximation algorithm $\mathcal{B}$ for some constant $c > 1$. Let $x \in I_P$ and some $\epsilon > 0$ be given. We claim that the algorithm returning the solution attaining $\text{goal}_P\{m(x, \mathcal{A}(x, \epsilon)), m(x, \mathcal{B}(x))\}$ combined with threshold function b defined as

$$b(1/\epsilon) = \begin{cases} 1 & \text{if } \epsilon \geq c - 1; \\ 1 + M_P(a(1/\epsilon)) & \text{otherwise,} \end{cases}$$

and $\epsilon_b := c - 1$ is a fiptas$^\infty$ for P.

Clearly, the function b is computable, because a and M_P are computable. As $\mathcal{A}$ and $\mathcal{B}$ run in time polynomial in $|x|$ for any instance $x \in I_P$, it remains to show that the returned solution is a $(1+\epsilon)$-approximate solution on instances $x \in I_P$ if $m^*(x) \geq b(1/\epsilon)$. If $\epsilon \geq \epsilon_b$, then $\mathcal{B}(x)$ ensures a feasible solution within c and thus within $(1+\epsilon)$ of $m^*(x)$. For $\epsilon < \epsilon_b$, a feasible solution is returned and if $m^*(x) \geq b(1/\epsilon) > M_P(a(1/\epsilon))$, then $|x| \geq a(1/\epsilon)$ by Lemma 2.3, assuring that $\mathcal{A}$ delivers a $(1+\epsilon)$-approximation. $\square$

Note that one can similarly prove that PTAS$^\omega \subseteq$ PTAS$^\infty$ and FPTAS$^\omega \subseteq$ FPTAS$^\infty$. However, we can also derive this from Proposition 5.11 and Theorem 3.6, respectively from Theorem 5.13, 5.14, and 3.4. Together with Theorem 5.12, this yields the following corollary.

Corollary 5.15 *EPTAS $\subseteq$ FIPTAS$^\infty$. The containment is strict, unless $P = NP$.*

This implies that the hierarchy of optimum-asymptotic approximation classes starts not only above FPTAS, but even above EPTAS (see Figure 1). It is an intriguing question whether this corollary can be strengthened to PTAS $\subseteq$ FPTAS$^\infty$, or whether PTAS $\not\subseteq$ FPTAS$^\infty$. We answer this question in Section 7 (Theorem 7.9).

6 Optimum-Asymptotic Schemes and Classic Classes

Asymptotic approximation schemes clearly play an important part in the hierarchy of approximation schemes. In the previous sections, we established inclusion and equivalence relations among the classes of problems admitting such schemes and more classic classes such as PTAS, EPTAS, and FPTAS. All inclusion relations are strict under some hardness condition. In some cases however, the hardness gap can be bridged. The next two sections build several of these bridges.

In this section, we give a new characterization of classic classes using optimum-asymptotic approximation schemes and concepts from fixed-parameter tractability. In this way, we can also prove that large classes of problems do not possess optimum-asymptotic schemes. Section 7 deals with asymptotic schemes in another way, in the sense that we try to increase the size or optimum of a problem instance to get around the threshold function of asymptotic schemes.

6.1 New Characterizations of Classic Classes

When we view optimum-asymptotic approximation schemes from the perspective of the theory of fixed-parameter tractability, we can obtain new characterizations of the classic classes of approximation schemes defined in Table 1. We first define some notions from fixed-parameter tractability, as found for instance in Downey and Fellows [17] and Flum and Grohe [18].

Definition 6.1 *In the* standard parameterization *(or* decision variant*) of a problem P, one is asked, given $x \in I_P$ and a positive integer k, to decide whether $m^*(x) \geq k$ if* $\text{goal}_P = \max$ *or* $m^*(x) \leq k$ *if* $\text{goal}_P = \min$.

Definition 6.2 ([38]) *A problem P is* simple *if its standard parameterization can be decided in time polynomial in $|x|$ for every instance $x \in I_P$ and every fixed k. It is* p-simple *if its standard parameterization can be decided in time polynomial in $|x|$ and k for every $x \in I_P$ and every k.*

Proposition 6.3 *The standard parameterization of a problem P belongs to the class XP if and only if P is simple. It belongs to the class PFPT (Polynomial FPT) if and only if P is p-simple.*

A precise definition of the classes PFPT and XP may be found in [8, 17, 18]. Here we only need an understanding of the restriction of these classes to standard parameterizations of optimization problems.

Definition 6.4 ([4, 6]) *An algorithm $\mathcal{A}$ decides the standard parameterization of a problem P* with witness *if $\mathcal{A}$ decides the standard parameterization of P and if it decides* Yes*, it also returns a $y \in S(x)$ such that $m(x, y) \geq k$ if* $\text{goal}_P = \max$ *or $m(x, y) \leq k$ if* $\text{goal}_P = \min$.

Using this definition, we can consider problems that are (p-)simple with witness and define classes XP^w and PFPT^w as expected.

We now give a new characterization of the classes PTAS and FPTAS.

Theorem 6.5 *A problem is*

- *in PTAS if and only if it has a ptas$^\infty$ and its standard parameterization is in XP^w.*
- *in FPTAS if and only if it has an fptas$^\infty$ with a polynomially-bounded threshold function and its standard parameterization is in PFPT^w.*

Proof: Consider a problem P and suppose that $P \in \text{PTAS}$. Then P is in PTAS^∞ by Proposition 5.11. It follows from a proof of Paz and Moran [38] that the standard parameterization of P is simple with witness (run the ptas with $\epsilon = 1/(k+1)$), and thus in XP^w.

For the converse, suppose that P is in PTAS^∞ and in XP^w. Let $\mathcal{A}$ be a ptas$^\infty$ for P with computable threshold function b and let $\mathcal{B}$ be an algorithm that decides the standard parameterization of P with witness in time polynomial in $|x|$ for every fixed k. Assume w.l.o.g. that $\text{goal}_P = \min$. The case when $\text{goal}_P = \max$ is similar.

Given an instance $x \in I_P$ and some $\epsilon > 0$, compute $b(1/\epsilon)$. For each integer $k \in [1, \ldots, b(1/\epsilon)]$, call $\mathcal{B}(x, k)$. If any of these calls returns a YES-answer, then $m^*(x)$ equals the smallest value of k for which $\mathcal{B}(x, k)$ gives a YES-answer. The witness solution $y \in S(x)$ returned by $\mathcal{B}$ in this case has $m(x, y) = m^*(x)$ and thus trivially is a $(1+\epsilon)$-approximation. If no call returns a YES-answer, then $m^*(x) \geq b(1/\epsilon)$ and $\mathcal{A}(x, \epsilon)$ returns a $(1+\epsilon)$-approximation to $m^*(x)$. In either case, we get a $(1+\epsilon)$-approximation.

The running time of this scheme is polynomial in $|x|$ for every fixed $\epsilon > 0$. For a fixed value of ϵ, $b(1/\epsilon)$ can be computed in constant time. Furthermore, $b(1/\epsilon)$ itself is a constant and hence $\mathcal{B}$ is called a constant number of times. Each call takes polynomial time. If none of these calls returns a YES-answer, we run $\mathcal{A}(x, \epsilon)$, which also takes polynomial time.

The proof of the characterization of FPTAS is similar. Since the threshold function is polynomially bounded, we may assume it is a polynomial. Since a polynomial can be evaluated in polynomial time, the theorem follows. □

The characterizations seem different from those given by Paz and Moran [38] and Chen et al. [8].

Example 6.6 *Jansen and Zhang [26] prove that the standard parameterization of Maximum Rectangle Packing (maximizing the number of given rectangles that can be packed in a given rectangle) is in XP^w. Then an fptas$^\infty$ for this problem is given, implying by Theorem 6.5 that it is in PTAS.*

Example 6.7 *Minimum Bin Packing is in FIPTAS$^\infty$ (see Example 5.2), but has no ptas unless P = NP [22]. Hence its standard parameterization is not in XP^w unless P = NP.*

A similar characterization can be given for the class EPTAS. Let *EPTAS$^\infty$* denote the class of problems admitting an eptas$^\infty$, i.e. a ptas$^\infty$ with running time $\text{poly}(|x|) \cdot f(1/\epsilon)$ for some computable function f. Call a problem *e-simple* if its standard parameterization can be decided in time $\text{poly}(|x|) \cdot f(k)$ for some computable function f. The standard parameterization of a problem belongs to FPT if and only if it is e-simple. Using Definition 6.4, we can define (similar to XP^w and PFPT^w) the class FPT^w.

Theorem 6.8 *A problem is in EPTAS if and only if it has an eptas$^\infty$ and its standard parameterization is in FPT^w.*

The proof is similar to the proof of Theorem 6.5.

6.2 Existence of Optimum-Asymptotic Approximation Schemes

Theorem 6.5 has interesting consequences. In particular, it gives the tools to improve on a theorem of Arora et al. [1]. They showed that MAX-3SAT has no ptas, unless P=NP. As a consequence, no MAX-SNP-complete problem can have a ptas, unless P=NP. We prove that this extends to ptas$^\infty$.

Definition 6.9 ([37]) *An NPO-problem P is in* MAX-NP *if it can be expressed as*

$$\max_S |\{\bar{a} \mid \exists \bar{b}\, \psi(\bar{a}, \bar{b}, G, S)\}|,$$

where G is an instance of P described as a finite structure, S ranges over all admissible structures, $\bar{a}$ and $\bar{b}$ are tuples of variables, and ψ is a quantifier-free boolean formula. Problem P is in MAX-SNP *if it can be expressed as*

$$\max_S |\{\bar{a} \mid \psi(\bar{a}, G, S)\}|,$$

with G, S, ψ, and $\bar{a}$ as before.

Kolaitis and Thakur [30] proved that MAX-SNP $\subset$ MAX-NP, as Maximum Satisfiability is in MAX-NP, but not in MAX-SNP.

We first need the following theorem, a weaker form of which was proved by Cai and Chen [4].

Theorem 6.10 *If P is in MAX-NP, then its standard parameterization is in FPT^w.*

Proof: Suppose that we are given an instance of P, described as a finite structure G, and a positive integer k. Because P is in MAX-NP, its instances can be expressed as

$$\max_S |\{\bar{a} \mid \exists \bar{b}\, \psi(\bar{a}, \bar{b}, G, S)\}|.$$

Since this expression is fixed irrespective of the instance, Papadimitriou and Yannakakis [37] showed that for a particular instance one only needs to consider a polynomial number of different values $\bar{a}$. Also, for every fixed value of $\bar{a}$, it again suffices to consider only a polynomial number of different values $\bar{b}$. Finally, for any fixed values of $\bar{a}$ and $\bar{b}$, let $\phi_{\bar{a},\bar{b}}(S) = \psi(\bar{a}, \bar{b}, G, S)$ be the resulting boolean formula after substituting $\bar{a}$, $\bar{b}$, and G. It consists of a constant number, say at most c, of variables of the form $Q(\bar{d})$, where Q is any predicate of S.

For every $\bar{a}$, let $B(\bar{a})$ denote the set of values $\bar{b}$ such that $\phi_{\bar{a},\bar{b}}(S)$ is satisfiable for some structure S. Let A denote the set of values $\bar{a}$ for which $B(\bar{a}) \neq \emptyset$. Then we can express P as:

$$\max_S |\{\bar{a} \in A \mid \phi_{\bar{a}}(S) := \textstyle\bigvee_{\bar{b} \in B(\bar{a})} \phi_{\bar{a},\bar{b}}(S)\}|.$$

Furthermore, let $n = |A|$ and $l = \max_{\bar{a} \in A} |B(\bar{a})|$. As both n and l are polynomial in the size of the input, $l \leq n^{O(1)}$.

Suppose that $k \leq n/2^c$. Papadimitriou and Yannakakis [37] proved that by fixing a particular $\bar{b}(\bar{a}) \in B(\bar{a})$ for each $\bar{a} \in A$, one can find in polynomial time a structure S satisfying at least $n/2^c$ clauses $\phi_{\bar{a},\bar{b}(\bar{a})}(S)$. Hence if $k \leq n/2^c$, the answer is trivially YES. Moreover, a witness to this can be found in polynomial time.

So suppose that $k > n/2^c$. Then we enumerate all assignments of variables of the form $Q(\bar{d})$ that occur (i.e. all relevant structures S) to verify whether the maximum number of satisfiable $\phi_{\bar{a}}(S)$ is at least k. There are at most n clauses $\phi_{\bar{a}}(S)$ which can be satisfied. For each such clause, if it is to be satisfied, there are at most l clauses $\phi_{\bar{a},\bar{b}}(S)$, from which we should choose one that must be satisfied. As each such clause $\phi_{\bar{a},\bar{b}}(S)$ consists of at most c variables of the form $Q(\bar{d})$, where Q is any predicate in S, we can enumerate all relevant structures S in $O(n \cdot (l+1)^n \cdot 2^{cn})$ time. Since $n < 2^c k$ and $l \leq n^{O(1)}$, this is $O(k^{O(k)})$ time. Therefore we can check in $O(k^{O(k)})$ time whether the instance of P has answer YES and, if so, return a witness structure S for this. □

Observe that for problems in MAX-SNP we can apply a similar proof as above, but with $l = 1$. Then the running time of the given algorithm improves to $O(k \cdot 2^{O(k)})$ plus a polynomial in the input size. Kratsch [33] recently showed that problems in MAX-NP admit a polynomial kernel, strengthening the above result.

Combining Theorem 6.10 with Theorem 6.5 and 6.8, we obtain the following result.

Theorem 6.11 *If a problem P is in MAX-NP and PTAS$^\infty$ (EPTAS$^\infty$), it is in PTAS (EPTAS).*

In the given form, the theorem gives a way to construct a ptas (eptas) for a problem in MAX-NP if the problem has a ptas$^\infty$ (eptas$^\infty$). Phrased differently however, it gives a powerful tool to prove that for some problems a ptas$^\infty$ (eptas$^\infty$) cannot exist.

Corollary 6.12 *If a problem P in MAX-NP cannot have a ptas (eptas) under some hardness condition, then it cannot have a ptas$^\infty$ (eptas$^\infty$) under the same hardness condition.*

This already proves the nonexistence of a ptas$^\infty$ for many problems, for instance for Maximum Satisfiability. However, a more general statement is possible. Arora et al. [1] showed that no MAX-SNP-complete problem can have a ptas, unless P=NP. We now strengthen this result as follows.

Theorem 6.13 *If a problem P is MAX-SNP-complete (under the L-reduction), then it cannot have a ptas$^\infty$, unless P=NP.*

This implies for instance that problems such as MAX-3SAT, Maximum Independent Set on bounded degree graphs, and Maximum Cut do not have a ptas$^\infty$, unless P=NP. In fact, using the result of Arora et al., one can even prove that for each MAX-SNP-complete problem P there is a fixed constant $c > 1$ such that P cannot be approximated (optimum-)asymptotically within c, unless P=NP.

It should be noted here that similar results can be proved for a syntactically defined class of minimization problems, called *MIN* $F^+\Pi_1$ [31], which includes Minimum Vertex Cover and many vertex-deletion and edge-deletion problems in graphs such as Minimum Feedback Arc Set. Cai and Chen [4] proved that the standard parameterizations of all problems in this class are in FPT^w. Hence we obtain the following theorem.

Theorem 6.14 *If a problem P is in MIN $F^+\Pi_1$ and in PTAS$^\infty$, then it is in PTAS.*

Similar to Corollary 6.12, one can use this theorem to prove negative results. For instance, Theorem 6.14 implies that Minimum Vertex Cover, which cannot have a ptas unless P=NP [37, 1], also cannot have a ptas$^\infty$ unless P=NP.

6.3 Approximation-Preserving Reductions

Due to results by Khanna et al. [29], we know that no APX-complete problem can have a ptas unless P=NP. Phrased differently, if for a problem P in APX there exists an approximation-preserving reduction from Maximum Satisfiability (or a specific bounded case of it) to P, then P cannot have a ptas unless P=NP. We prove that a similar statement can be made about ptas$^\infty$ by using a different type of approximation-preserving reduction.

The result of Khanna et al. holds under the PTAS-reduction, defined by Crescenzi and Trevisan [15].

Definition 6.15 *There is a* PTAS-reduction *from a problem P to a problem P' if there exist computable functions t_1, t_2, and $c : \mathbb{Q}^+ \to \mathbb{Q}^+$ such that for any $x \in I_P$ and any $\epsilon > 0$,*

1. *$t_1(x, \epsilon) \in I_{P'}$ and $t_1(x, \epsilon)$ is computable in time polynomial in $|x|$ for any fixed value of ϵ;*
2. *for any $y \in S_{P'}(t_1(x, \epsilon))$, $t_2(x, y, \epsilon) \in S_P(x)$ and $t_2(x, y, \epsilon)$ is computable in time polynomial in $|x|$ and $|y|$ for any fixed value of ϵ;*
3. *for any $y \in S_{P'}(t_1(x, \epsilon))$, if y is within $1 + c(\epsilon)$ of $m^*_{P'}(t_1(x, \epsilon))$, then $t_2(x, y, \epsilon)$ is within $1 + \epsilon$ of $m^*_P(x)$.*

Several well-known reductions are a special case of PTAS-reductions, such as P-reductions [36], L-reductions [37], E-reductions [29], and AP-reductions [13]. The most important property of all these reductions is that they preserve membership of PTAS.

Lemma 6.16 ([15]) *If there is a PTAS-reduction from P to P' and P' has a ptas, then P also has a ptas.*

Proof: Let $\mathcal{A}'$ be a ptas for P' and (t_1, t_2, c) a PTAS-reduction from P to P'. It can be easily seen that given $x \in I_P$ and some $\epsilon > 0$, computing $t_2(x, \mathcal{A}'(t_1(x, \epsilon), c(\epsilon)), \epsilon)$ yields a ptas for P. □

Most PTAS-reductions given in the literature actually also preserve membership of PTAS$^\infty$. This is due to the following property.

Lemma 6.17 *Suppose there is a PTAS-reduction (t_1, t_2, c) from P to P' and a monotone computable function $f : \mathbb{N} \to \mathbb{N}$ with $\liminf_{n\to\infty} f(n) = \infty$ such that for any $\epsilon > 0$ and any $x \in I_P$, $m^*_{P'}(t_1(x, \epsilon)) \geq f(m^*_P(x))$. Then if P' has a ptas$^\infty$, P also has a ptas$^\infty$.*

Proof: Suppose that P' has a ptas$^\infty$ $\mathcal{A}'$ such that any instance $x' \in I_{P'}$ can be approximated within $(1+\epsilon)$ if $m^*(x') \geq b'(1/\epsilon)$ for some computable function b'. Let $x \in I_P$ and some $\epsilon > 0$ be given. We claim that computing $\mathcal{A}(x,\epsilon) = t_2(x, \mathcal{A}'(t_1(x,\epsilon), c(\epsilon)), \epsilon)$ yields a ptas$^\infty$ for P with some suitably chosen threshold function b.

It follows from the proof of Lemma 6.16 that $\mathcal{A}(x,\epsilon)$ always returns a feasible solution y and that y is within $(1+\epsilon)$ of $m^*_P(x)$ if $m^*_{P'}(t_1(x,\epsilon)) \geq b'(1/c(\epsilon))$. Hence to prove that $\mathcal{A}$ is a ptas$^\infty$ we need a computable function b such that $m^*_P(x) \geq b(1/\epsilon)$ implies that $m^*_{P'}(t_1(x,\epsilon)) \geq b'(1/c(\epsilon))$. Choosing

$$b(1/\epsilon) = \min\{n \mid f(n) \geq b'(1/c(\epsilon))\},$$

which is a computable function, we get

$$m^*_P(x) \;\geq\; b(1/\epsilon) \quad\Rightarrow\quad f(m^*_P(x)) \;\geq\; f(b(1/\epsilon)) \quad\Rightarrow\quad m^*_{P'}(t_1(x,\epsilon)) \;\geq\; b'(1/c(\epsilon)),$$

proving the lemma. □

The lemma proves the usefulness of the following notion.

Definition 6.18 *There is a* PTAS$^\infty$-reduction *from a problem P to a problem P' if there is a PTAS-reduction (t_1, t_2, c) from P to P' and a monotone computable function $f : \mathbb{N} \to \mathbb{N}$ with $\liminf_{n\to\infty} f(n) = \infty$ such that for any $\epsilon > 0$ and any $x \in I_P$, $m^*_{P'}(t_1(x,\epsilon)) \geq f(m^*_P(x))$.*

Observe that the PTAS$^\infty$-reduction is transitive. Moreover, by Lemma 6.17, PTAS$^\infty$-reductions preserve membership of PTAS$^\infty$.

Some reductions appearing in the literature are a special case of PTAS$^\infty$-reductions, such as asymptotic continuous reductions [41] and (polynomial time) ratio-preserving reductions [38]. However, they are rather restrictive and not many have been shown to exist. Here we will rely on Lemma 6.17 instead.

A first question we need to answer with respect to PTAS$^\infty$-reductions is whether there exist any (natural) problems that are APX-complete under the PTAS$^\infty$-reduction. Such a problem indeed exists. Let Maximum Bounded Weighted Satisfiability be the variant of Maximum Satisfiability in which each variable x_i has weight w_i such that $W \leq \sum_i w_i \leq 2W$ for some given weight W.

Theorem 6.19 *Maximum Bounded Weighted Satisfiability (MBWS) is APX-complete under the PTAS$^\infty$-reduction.*

Proof: Crescenzi and Panconesi [14] proved that MBWS is APX-complete under the P-reduction. Upon closer inspection and using Definition 6.18, it can be seen that the given reductions are also PTAS$^\infty$-reductions. □

Using this theorem, we can in fact prove that there is a problem in MAX-SNP that is APX-complete under the PTAS$^\infty$-reduction.

Theorem 6.20 *Maximum 3-Satisfiability is APX-complete under the PTAS$^\infty$-reduction.*

Proof: Crescenzi and Trevisan [15] presented a PTAS-reduction from MBWS to a polynomially-bounded variant of it, Maximum Polynomially-Bounded Weighted Satisfiability. Khanna et al. [29] showed that any polynomially-bounded problem in APX has an E-reduction to Maximum 3-Satisfiability. Upon closer inspection of these reductions, one can see that they are in fact PTAS$^\infty$-reductions. □

Observe that any problem that is APX-complete under the PTAS$^\infty$-reduction cannot have a ptas$^\infty$, unless P=NP.

Lemma 6.21 *If a problem P is APX-complete under the PTAS$^\infty$-reduction, then it cannot have a ptas$^\infty$, unless P=NP.*

Proof: We established in Theorem 6.13 that no MAX-SNP-complete problem can have a ptas$^\infty$ unless P=NP. In particular, Maximum 3-Satisfiability, which is APX-complete, has no ptas$^\infty$. Let P be an APX-complete problem under the PTAS$^\infty$-reduction. If P had a ptas$^\infty$, then by the APX-completeness of P, Maximum 3-Satisfiability also has a ptas$^\infty$. This is a contradiction, unless P=NP. $\square$

It is interesting to note that several reductions proving that no ptas$^\infty$ can exist for a certain problem, actually use a PTAS$^\infty$-reduction implicitly. For example, a result of Woeginger [46] showing that Minimum 2-Dimensional Vector Packing cannot have a ptas$^\infty$ unless P=NP can be explained this way. In Minimum 2-Dimensional Vector Packing, we want to partition a given set of vectors in $[0,1] \times [0,1]$ into a minimum number of subsets, such that in every subset the sum of all vectors is at most 1 in every coordinate.

Theorem 6.22 *Minimum 2-D Vector Packing is APX-complete under the PTAS$^\infty$-reduction. Hence it cannot have a ptas$^\infty$ unless P=NP.*

Proof: Consider the following problems. In Maximum 3-Dimensional Matching, we are given three sets X, Y, and Z, each of size q, and a set $T \subseteq X \times Y \times Z$ of triples, and we are asked to find the maximum number of triples that do not agree on any coordinate. In Maximum Bounded 3-Dimensional Matching, we additionally impose that each element occurs in at least one, but at most three triples. Kann [27] gives an L-reduction to Maximum Bounded 3-D Matching from Maximum 3-Satisfiability-B, which itself has an L-reduction from Maximum 3-Satisfiability [37]. Both L-reductions are actually also PTAS$^\infty$-reductions.

The construction of Woeginger [46] reduces instances x of Maximum Bounded 3-Dimensional Matching to instances x' of Minimum 2-Dimensional Vector Packing. From the construction, x has a solution of value at least k if and only if x' has a solution of value at most $q + \lceil \frac{1}{3}(|T| - k) \rceil$. This leads to a PTAS$^\infty$-reduction as follows. For any $\epsilon > 0$, for any $y' \in S(x')$ with $m(x', y') \leq (1+\epsilon) \cdot m^*(x')$, and for $y = t_2(x', y', \epsilon)$,

$$q + \tfrac{1}{3}(|T| - m(x,y)) \;=\; m(x',y') \;\leq\; (1+\epsilon) \cdot \left(q + \left\lceil \tfrac{1}{3}(|T| - m^*(x)) \right\rceil\right).$$

Then

$$m(x,y) \;\geq\; m^*(x) - 3\epsilon \cdot \left(q + \left\lceil \tfrac{1}{3}\left(|T| - m^*(x)\right) \right\rceil\right) \;\geq\; (1 - O(\epsilon)) \cdot m^*(x),$$

since $q/7 \leq m^*(x) \leq 3q$ from the definition of Maximum Bounded 3-Dimensional Matching. Moreover, $m^*(x') \geq q$, and thus $m^*(x') \geq m^*(x)/3$. This yields the desired PTAS$^\infty$-reduction. $\square$

We can also be interested in the existence of PTAS$^\infty$-reductions. Crescenzi et al. [13] showed that Minimum Bin Packing is not APX-complete under the AP-reduction (and thus also under the PTAS-reduction), unless the polynomial hierarchy collapses. Furthermore, they remark that this result "does not seem to be obtainable" under the condition that P=NP. The reason for this is that Crescenzi et al. show that if NP=co-NP, then there is an AP-reduction from Maximum Satisfiability to Minimum Bin Packing. Thus NP=co-NP would imply that P=NP, which is highly unlikely. If however we consider PTAS$^\infty$-reductions, then the result of Crescenzi et al. can be obtained under the condition that P=NP.

Theorem 6.23 *Minimum Bin Packing is not APX-complete under the PTAS$^\infty$-reduction, unless P=NP.*

Proof: Recall that Minimum Bin Packing has a ptas$^\infty$ (see Example 5.2). Hence if Minimum Bin Packing were APX-complete under the PTAS$^\infty$-reduction, then this would contradict Lemma 6.21, unless P=NP. $\square$

In particular, the theorem implies that no PTAS$^\infty$-reduction can exist from Maximum Satisfiability to Minimum Bin Packing, unless P=NP. It should be noted here that the AP-reduction given by Crescenzi et al. under the condition that NP=co-NP is not a PTAS$^\infty$-reduction, since it has $m^*(t_1(x,\epsilon)) = O(1)$ for any instance x of Maximum Satisfiability and any $\epsilon > 0$. Therefore Theorem 6.23 does not contradict, but augments the results of Crescenzi et al. [13]. Furthermore, it can be easily seen that Theorem 6.23 extends to several other problems, including Minimum Degree Spanning Tree and Chromatic Index.

7 Pumpable Problems

Looking back at the previous sections, one can notice that in the equivalence proofs we do not always get the equivalence we hope for. For instance, one expects an fptas$^\omega$ with $a(1/\epsilon) = 2^{1/\epsilon}$ to be equivalent to an eptas with $f(1/\epsilon) = 2^{1/\epsilon}$. However, this does not seem to hold in general, as the proof of Theorem 3.4 only gives $f(1/\epsilon) = 2^{\text{poly}(a(1/\epsilon))}$. Hence we are interested in properties of problems for which the equivalences *are* nice. Similarly, we want to know if for certain types of problems the hierarchy developed in the previous sections collapses. A promising property of problems seems to be pumpability.

Definition 7.1 *An optimization problem P is k-*pumpable *if there exist functions g_1 and g_2 such that for any instance $x \in I_P$*

- *$g_1(x) \in I_P$ and $S_P(g_1(x)) \neq \emptyset$ if $S_P(x) \neq \emptyset$;*
- *for every $r \geq 1$ and for every $y \in S_P(g_1(x))$ within a factor r of $m^*(g_1(x))$, $g_2(y) \in S_P(x)$ and $g_2(y)$ is within r of $m^*(x)$;*
- *g_1 and g_2 are computable in time polynomial in $|x|$ and $\log k$;*
- *some* pumpability condition *holds.*

Note that in the logarithmic cost model we use, it makes sense that the running times of g_1 and g_2 depend polynomially on $\log k$.

We distinguish two pumpability conditions, one related to the size of $g_1(x)$ and one related to the optimal objective value of $g_1(x)$.

Definition 7.2 *An optimization problem P is k-*size-pumpable *if P is k-pumpable with the condition that $|g_1(x)| \geq |x| + k$. An optimization problem P is k-*opt-pumpable *if P is k-pumpable with the condition that $m^*(g_1(x)) \geq m^*(x) + k$.*

It appears that many problems possess either one of these properties or even both, as can be seen in the following example.

Example 7.3 *Minimum Vertex Cover is 1-size-pumpable, because one can take for g_1 the function that makes two disjoint copies of an instance. The desired property of the translation back follows by the pigeonhole principle. Using the same idea, Minimum Vertex Cover is also 1-opt-pumpable. Minimum Makespan Scheduling is k-opt-pumpable for any k by multiplying all job lengths by $k+1$. Using a similar idea it is also 1-size-pumpable.*

Observe that problems that are k-size-pumpable for arbitrary values of k cannot exist. Otherwise one could take $k = 2^{\text{poly}(|x|)}$ and thus add an exponential number of bits to an instance x of P in time polynomial in $|x|$.

We consider the question which problems are pumpable in more detail in Paragraph 7.3. First, we show how the property of being pumpable helps to prove some new equivalences among the classes we defined previously. In the following, when we talk about a size- or opt-pumpable problem, we assume that the functions g_1 and g_2 are known.

7.1 Optimum-Asymptotic Schemes and Pumpability

It follows from Theorem 5.12 that PTAS $\subset$ PTAS$^\infty$, unless P=NP. For 1-opt-pumpable problems however, the two classes are equivalent.

Lemma 7.4 *Let P be 1-opt-pumpable and in PTAS$^\infty$. Then $P \in$ PTAS.*

Proof: Assume we have a ptas$^\infty$ $\mathcal{A}$ for P with computable threshold function b. Given $x \in I_P$ and some $\epsilon > 0$, compute $b(1/\epsilon)$, which takes constant time for any fixed ϵ. Then pump x $b(1/\epsilon)$ times to get an instance x'. Note that the size of the output of g_1 is polynomial in the size of the input. Hence pumping $b(1/\epsilon)$ times means that x' has size at most $|x|^{O(1)^{b(1/\epsilon)}}$ and thus the pumping steps can be done in time polynomial in $|x|$ for every fixed ϵ.

As x has been pumped $b(1/\epsilon)$ times, $m^*(x') \geq b(1/\epsilon) + m^*(x) \geq b(1/\epsilon)$. Hence we can compute $y' = \mathcal{A}(x', \epsilon)$ and by the definition of ptas$^\infty$, y' is within $1 + \epsilon$ of $m^*(x')$. Furthermore, y' can be computed in time polynomial in $|x|$ for every fixed ϵ, as $|x'|$ is polynomial in $|x|$ for every fixed ϵ. Iteratively applying g_2 to y', we get a solution y for x within $1 + \epsilon$ of $m^*(x)$. As we need to apply g_2 only $b(1/\epsilon)$ times, this again takes time polynomial in $|x|$ for every fixed ϵ. □

There are several ways in which one could use this lemma. First of all, it provides a condition under which problems are not 1-opt-pumpable.

Theorem 7.5 *Any problem that is in PTAS$^\infty$ but not in PTAS (unless P=NP), is not* 1*-opt-pumpable (unless P=NP).*

This is an immediate consequence of Lemma 7.4. There are several examples of problems that fit the theorem.

Corollary 7.6 *Minimum Bin Packing, Chromatic Index, and Minimum Degree Spanning Tree are not* 1*-opt-pumpable, unless P=NP.*

Secondly, Lemma 7.4 shows that some problems cannot have a ptas$^\infty$. Consider for instance the variation of Minimum Bin Packing with precedence constraints. (The precedence constraints state that for certain items i and j, item i has to appear in a bin with a lower number than item j.) It cannot have a ptas, as Minimum Bin Packing itself cannot have a ptas unless P=NP [21]. Queyranne [40] already proved the following result. But as Minimum Bin Packing with Precedence is 1-opt-pumpable (make two copies of the instance and by usung precedence constraints ensure that items of the first instance have to come before items of the second), it now follows as a corollary of Lemma 7.4.

Theorem 7.7 *Minimum Bin Packing with Precedence has no ptas$^\infty$, unless P=NP.*

Actually, Queyranne applied a similar form of pumping to the problem in order to obtain his result.

We now consider the effect of pumpability on problems in FPTAS$^\infty$.

Lemma 7.8 *Let P be* 1*-opt-pumpable and in FPTAS$^\infty$ and let $c > 0$ be a constant. If $|g_1(x)| \leq c \cdot |x|$ for any $x \in I_P$, then $P \in$ EPTAS.*

Proof: Following the proof of Lemma 7.4, the condition on g_1 now implies that after pumping $b(1/\epsilon)$ times we obtain an instance of size $O(c^{b(1/\epsilon)} \cdot |x|)$. Then the construction of Lemma 7.4 actually takes time bounded by a polynomial in $|x|$ times some (computable) function of $1/\epsilon$. □

Using this lemma, one can prove interesting results about the relation of FPTAS$^\infty$ to PTAS$^\infty$ and PTAS.

Theorem 7.9 *PTAS $\not\subseteq$ FPTAS$^\infty$, unless FPT=W[1].*

Proof: Let P denote the minimum dominating set problem in unit disk graphs. Hunt et al. [25] showed that $P \in$ PTAS. Suppose that $P \in$ FPTAS$^\infty$. Because P is easily 1-opt-pumpable with a linear size output (let g_1 just take two disjoint copies of the graph of the instance), it has an EPTAS by Lemma 7.8. But then P is in FPT (with respect to its standard parameterization) by results of Bazgan [3] and Cesati and Trevisan [7]. However, as P is W[1]-hard [34], this is not possible, unless FPT=W[1]. □

This leads to the following corollary, which we could not derive yet in Section 5.

Corollary 7.10 *PTAS$^\infty \neq$ FPTAS$^\infty$, unless FPT=W[1].*

Although Lemma 7.8 gives a way to go from an fptas$^\infty$ to an eptas, it only holds if the output of g_1 has linear size. If that is not the case, the following lemma can be useful.

Lemma 7.11 *Let P be k-opt-pumpable for any k and in FPTAS$^\infty$. Then $P \in$ EPTAS.*

Proof: In this case it suffices to pump once with $k = b(1/\epsilon)$ to make the construction of Lemma 7.4 work. □

If additionally $b(1/\epsilon)$ is bounded by $2^{\text{poly}(1/\epsilon)}$, then $P \in$ FPTAS, as both g_1 and g_2 can be computed in time polynomial in $|x|$ and $\log k = \log b(1/\epsilon) = \text{poly}(1/\epsilon)$. This last observation has interesting consequences.

Example 7.12 *Extensible Bin Packing (where bin size is part of the input) and Minimum Makespan Scheduling are strongly NP-hard, polynomially-bounded problems and thus have no fptas unless P=NP (see Corollary 4.13). However, both problems are k-opt-pumpable for any k (multiply all numbers of the instance by k). Hence by Lemma 7.11 they cannot have a fptas$^\infty$ where $b(1/\epsilon)$ is bounded by $2^{\text{poly}(1/\epsilon)}$, unless P=NP.*

For these two problems, the above facts were already known by results of Coffman and Lueker [12] and (in a weaker form) of Hochbaum and Schmoys [23], but here they are just a consequence of the general statement in Lemma 7.11. Moreover, the result of Hochbaum and Schmoys for Minimum Makespan Scheduling is strengthened by it.

7.2 Size-Asymptotic and Convergent Schemes and Pumpability

For size-asymptotic problems, the situation is slightly different. Recall that PTAS$^\omega$ and PTAS are equivalent for all problems, not just for pumpable problems (Theorem 3.6). The classes FPTAS$^\omega$ and EPTAS are also equivalent (Theorem 3.4), but turning an fptas$^\omega$ with threshold function a into an eptas currently increases the time complexity by at least a factor $2^{\text{poly}(a(1/\epsilon))}$. This is rather unfortunate. If a problem is 1-size-pumpable however, this exponential increase is not necessary.

Lemma 7.13 *Let P be 1-size-pumpable and in FPTAS$^\omega$ with computable threshold function a. Then $P \in$ EPTAS with running time polynomial in $|x|$, $1/\epsilon$, $a(1/\epsilon)$, and the time needed to compute $a(1/\epsilon)$.*

Proof: Assume that P has an fptas$^\omega$ $\mathcal{A}$ with threshold function a. Given some $x \in I_P$ and $\epsilon > 0$, compute $a(1/\epsilon)$. Pump x the smallest number of times needed to get an instance x' with $|x'| \geq a(1/\epsilon)$. Note that x' has size at most polynomial in $a(1/\epsilon)$ and $|x|$. Hence computing $y' = \mathcal{A}(x', \epsilon)$ takes time polynomial in $1/\epsilon$, $a(1/\epsilon)$, and $|x|$. Repeatedly applying g_2 to y' also takes time polynomial in $a(1/\epsilon)$ and $|x|$ and yields a solution to x within $(1+\epsilon)$ of $m^*(x)$. □

If $a(1/\epsilon)$ is computable in time polynomial in $1/\epsilon$ and $a(1/\epsilon)$, then the exponential increase is avoided. In particular, if the threshold function a of the fptas$^\omega$ is a polynomial, then $P \in$ FPTAS.

We can also show that for 1-size-pumpable problems the classes PCONV[f] and PTAS[f] coincide for many functions f, such as the logarithm.

Lemma 7.14 *Let $f \in \mathcal{F}^*$ be such that $f(n \cdot m) \geq f(n) + f(m)$. If $P \in$ PCONV[f] is 1-size-pumpable, then $P \in$ PTAS[f].*

Proof: Suppose that $P \in \text{PCONV}[f]$ and let $\mathcal{A}$ be a pconv$[f]$ for P. Consider an arbitrary $x \in I_P$ and some (fixed) $\epsilon > 0$. Let $\alpha \geq 1$ be some integer to be chosen later. Pump x the smallest number of times needed to get an instance x' with $|x'| \geq |x|^\alpha$ and run $\mathcal{A}(x')$. This yields a $(1+1/f(|x'|))$-approximation. We claim that α can be chosen such that $1+1/f(|x'|) \leq 1+\epsilon/f(|x|)$ and that we thus have bootstrapped $\mathcal{A}$ to a ptas$[f]$. The claim holds if

$$\frac{1}{f(|x'|)} \leq \frac{1}{f(|x|^\alpha)} \leq \frac{\epsilon}{f(|x|)}.$$

The first inequality is true by the monotonicity of f and the definition of x'. For the second inequality, note that since $f(n \cdot m) \geq f(n) + f(m)$,

$$\frac{1}{f(|x|^\alpha)} \leq \frac{1}{\alpha \cdot f(|x|)}.$$

As ϵ is fixed, we can choose any $\alpha \geq 1/\epsilon$ to ensure that $1/f(|x|^\alpha) \leq \epsilon/f(|x|)$. □

In Example 4.3 we noted that Maximum Independent Set on single-crossing-minor-free graphs has a pconv$[\log|x|]$. By the above lemma, we conclude that it also has a ptas$[\log|x|]$.

Pumpability also leads to several negative results.

Lemma 7.15 *Let P be an NP-hard, polynomially-bounded optimization problem with p the corresponding bounding polynomial plus 1. If P is 1-size-pumpable, then for any constant $\alpha > 0$, $P \notin$ PCONV$[p(|x|)^\alpha]$, unless P=NP.*

Proof: We may assume that p is a monotone polynomial. Suppose by way of contradiction that $P \in \text{PCONV}[p(|x|^\alpha)]$. Then there exists an algorithm $\mathcal{A}$ such that for any instance x of P, $\mathcal{A}(x)$ runs in time polynomial in $|x|$ and delivers a solution with approximation ratio $1 + 1/p(|x|)^\alpha$. Consider an arbitrary $x \in I_P$. Pump x the smallest number of times (say k times) needed to get an instance x' with $|x'| \geq p(|x|)^{1/\alpha}$. As k is bounded by a polynomial, pumping takes polynomial time. Let $y' = \mathcal{A}(x')$ and let $y \in S_P(x)$ be the resulting solution after iteratively applying g_2 to y' for k times. Observe that if P is a minimization problem, then

$$m(x', y') \leq (1 + 1/p(|x'|)^\alpha) \cdot m^*(x') \leq (1 + 1/p(|x|^{1/\alpha})^\alpha) \cdot m^*(x') \leq (1 + 1/p(|x|)) \cdot m^*(x')$$

and thus

$$m(x, y) \leq (1 + 1/p(|x|)) \cdot m^*(x).$$

If P is a maximization problem, we similarly obtain that $m^*(x) \leq (1 + 1/p(|x|)) \cdot m(x, y)$. This means that we found a pconv$[p]$, which is not possible by Lemma 4.12, unless P=NP. □

This lemma allows a simple subdivision of the PCONV-hierarchy.

Lemma 7.16 *PCONV$[|x|^\alpha] \subset$ PCONV$[\log|x|]$ for any (fixed) $\alpha > 0$, unless P=NP.*

Proof: Maximum Independent Set on single-crossing-minor-free graphs is in PCONV$[\log|x|]$ (see Example 4.3), but is also polynomially-bounded and 1-size-pumpable. □

This result can be strengthened significantly though. Huang [24] showed that Minimum Vertex Cover on planar graphs cannot have an eptas with $f(1/\epsilon) = 2^{o(\sqrt{1/\epsilon})}$, unless FPT=W[1]. Marx [35] proved that even an eptas with $f(1/\epsilon) = 2^{O((1/\epsilon)^{1-\delta})}$ for some $\delta > 0$ cannot exist, unless n-variable 3-SAT can be solved in $2^{o(n)}$ time.

Lemma 7.17 *PCONV$[\log^2|x|] \subset$ PCONV$[\log|x|]$, unless FPT=W[1]. PCONV$[\log^{1/(1-\delta)}|x|] \subset$ PCONV$[\log|x|]$ for any $\delta > 0$, unless n-variable 3-SAT can be solved in $2^{o(n)}$ time.*

7.3 Which Problems Are Pumpable?

We already showed that several problems are pumpable (e.g. Minimum Vertex Cover and Minimum Makespan Scheduling). In Theorem 7.5, we proved that several other problems (such as Minimum Bin Packing) are not 1-opt-pumpable, and it seems unlikely that they are 1-size-pumpable. In fact, many problems seem both 1-size-pumpable and 1-opt-pumpable, or both not. We give some evidence why this might not be a coincidence. At the moment, we do not know of any problem which provably possesses only one of the properties and not both.

To prove the pumpability of several classes of problems, we consider problems that are m^*-*opt-pumpable*. Essentially, this means that we can pump to (at least) double the objective value of the optimum. For any problem P and any $x \in I_P$, note that because $m^*(x) \leq M_P(|x|)$, $\log m^*(x) \leq \log M_P(|x|) \leq \text{poly}(|x|)$. Hence for any m^*-opt-pumpable problem the functions g_1 and g_2 are computable in time polynomial in $|x|$.

Lemma 7.18 *Let P be m^*-opt-pumpable. Then P is* 1*-size-pumpable.*

Proof: Given an instance $x \in I_P$, repeatedly opt-pump x to an instance x' until $|x'| > |x|$. We claim it takes at most time polynomial in $|x|$ until such an x' is found. As a first step, we prove that we only need to pump a polynomial amount of times. By Lemma 2.3, if $m^*(x') > M_P(|x|) = 2^{r_P(|x|,q_P(|x|))}$, then $|x'| > |x|$. As m^*-opt-pumping (at least) doubles the objective value of the optimum, pumping $1 + r_P(|x|, q_P(|x|))$ times gives an x' with $|x'| > |x|$. Second, note that before any pumping step, the size of the instance is at most $|x|$. Hence any pumping step costs only time polynomial in $|x|$. Therefore all needed pumping steps can be done in time polynomial in $|x|$ and thus P is 1-size-pumpable. □

Many graph optimization problems, such as Minimum Vertex Cover and Maximum Independent Set, are m^*-opt-pumpable and thus 1-size-pumpable. But when is a problem m^*-opt-pumpable?

Consider the following property of optimization problems.

Definition 7.19 ([29]) *A problem P is* additive *if there exists an operator $+$ and a polynomial-time computable function f such that $+$ maps any pair of instances $x_1, x_2 \in I_P$ to an instance $x_1 + x_2 \in I_P$ such that $m^*(x_1 + x_2) = m^*(x_1) + m^*(x_2)$ and f maps a (feasible) solution y to $x_1 + x_2$ to a pair of (feasible) solutions y_1, y_2 to x_1 and x_2 respectively such that $m(x_1 + x_2, y) = m(x_1, y_1) + m(x_2, y_2)$.*

This notion is similar to the notion of *paddable optimization problems* [9, 39].

From the definitions of additive and pumpable and Lemma 7.18, one can easily prove the following theorem.

Theorem 7.20 *Any additive problem is m^*-opt-pumpable and hence* 1*-opt-pumpable and* 1*-size-pumpable.*

Khanna et al. [29] remark that many problems are additive, such as Maximum Clique, Chromatic Number, Minimum Set Cover, and all problems in the class MAX-SNP.

Corollary 7.21 *Any problem in MAX-SNP is both* 1*-opt-pumpable and* 1*-size-pumpable.*

However, there also problems that are not (easily seen to be) additive, but that are m^*-opt-pumpable, such as Maximum Knapsack and Longest Path.

When we combine Lemma 7.4 with Corollary 7.21, we obtain the following weaker version of Theorem 6.11: If a problem P is in MAX-SNP and in PTAS$^\infty$, then P is in PTAS. In this way, most results from Paragraph 6.2 also follow by using pumpability.

8 Conclusion and Open Problems

In this paper, we defined several new types of approximation schemes and uncovered many interesting new relationships between classes of problems that can be approximated using these schemes and existing approximation classes. In particular, we have shown that EPTAS is a central class in the landscape of approximation classes. We also mapped the entire hierarchy of these classes, shown in Figure 1.

There are several intriguing questions left. The notion of pumpability, introduced in Section 7, gives a possibility for bridging the gap between optimum-asymptotic schemes and nonasymptotic schemes. But we have very few properties to check whether a problem is pumpable. Can size- or opt-pumpable problems be characterized? Another interesting question is whether every opt-pumpable problem is also size-pumpable and vice versa. We gave some evidence in Lemma 7.18 why one direction might be true, but it goes too far to conjecture that it holds both ways.

Convergent approximation schemes also pose new challenges. For the class PCONV[$\mathcal{F}^*$], we know that it is equivalent to EPTAS. However, the general classes PTAS[$\mathcal{F}^*$] and FPTAS[$\mathcal{F}^*$] remain mysterious. We know that some problems in EPTAS also lie in PTAS[$\log n$] (using Theorem 4.15) and that FPTAS=FPTAS[$\mathcal{P}$] (Theorem 4.10), but beyond this it seems hard to make conjectures about these classes.

Acknowledgements

The authors would like to thank Lex Schrijver for several helpful suggestions.

References

[1] Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M., "Proof Verification and the Hardness of Approximation Problems", *Journal of the ACM* 45:3 (May 1998), pp. 501–555.

[2] Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M., *Complexity and Approximation – Combinatorial Optimization Problems and Their Approximation Properties*, Springer-Verlag, Berlin, 1999.

[3] Bazgan, C., *Schémas d'approximation et complexité paramétrée*, Rapport de stage de DEA d'Informatique, Université Paris-Sud, Orsay, 1995.

[4] Cai, L., Chen, J., "On Fixed-Parameter Complexity and Approximability of NP Optimization Problems", *Journal of Computer and System Sciences* 54:3 (June 1997), pp. 465–474.

[5] Cai, L., Fellows, M., Juedes, D., Rosamond, F., "The Complexity of Polynomial-Time Approximation", *Theory of Computing Systems* 41:3 (October 2007), pp. 459–477.

[6] Cai, L., Juedes, D., "On the Existence of Subexponential Parameterized Algorithms", *Journal of Computer and System Sciences* 67:4 (December 2003), pp. 789–807.

[7] Cesati, M., Trevisan, L., "On the Efficiency of Polynomial Time Approximation Schemes", *Information Processing letters* 64:4 (November 1997), pp. 165–171.

[8] Chen, J., Huang, X., Kanj, I.A., Xia, G., "Polynomial Time Approximation Schemes and Parameterized Complexity", *Discrete Applied Mathematics* 155:2 (January 2007), pp. 180–193.

[9] Chen, Z.-Z., Toda, S., "On the Complexity of Computing Optimal Solutions", *International Journal of Foundations of Computer Science* 2:3 (September 1991), pp. 207–220.

[10] Chiba, N., Nishizeki, T., Saito, N., "Applications of the Lipton and Tarjan's Planar Separator Theorem", *Journal of Information Processing* 4:4 (1981), pp. 203–207.

[11] Coffman, Jr., E.G., Garey, M.G., Johnson, D.S., "Approximation Algorithms for Bin Packing: A Survey" in Hochbaum, D.S. (ed.) *Approximation Algorithms for NP-Hard Problems*, PWS Publishing Company, Boston, 1997, pp. 46–93.

[12] Coffman, Jr., E.G, Lueker, G.S., "Approximation Algorithms for Extensible Bin Packing", *Journal of Scheduling* 9:1 (February 2006), pp. 63–69.

[13] Crescenzi, P., Kann, V., Silvestre, R., Trevisan, L., "Structure in Approximation Classes", *SIAM Journal on Computing* 28:5 (1999), pp. 1759–1782.

[14] Crescenzi, P., Panconesi, A., "Completeness in Approximation Classes", *Information and Computation* 93:2 (August 1991), pp. 241–262.

[15] Crescenzi, P., Trevisan, L., "On Approximation Scheme Preserving Reducibility and its Applications", *Theory of Computing Systems* 33:1 (June 2000), pp. 1–16.

[16] Demaine, E.D., Hajiaghayi, M., Nishimura, N., Ragde, P., Thilikos, D.M., "Approximation Algorithms for Classes of Graphs Excluding Single-Crossing Graphs as Minors", *Journal of Computer and System Sciences* 69:2 (September 2004), pp. 166–195.

[17] Downey, R.G., Fellows, M.R., *Parameterized Complexity*, Springer-Verlag, New York, 1999.

[18] Flum, J., Grohe, M., *Parameterized Complexity Theory*, Springer-Verlag, Berlin, 2006.

[19] Fürer, M., Raghavachari, B., "Approximating the Minimum-Degree Steiner Tree to Within One of Optimal", *Journal of Algorithms* 17:3 (November 1994), pp. 409–423.

[20] Garey, M.R., Johnson, D.S., "The Complexity of Near-Optimal Graph Coloring", *Journal of the ACM* 23:1 (January 1976), pp. 43–49.

[21] Garey, M.R., Johnson, D.S., "'Strong' NP-Completeness Results: Motivation, Examples, and Implications", *Journal of the ACM* 25:3 (July 1976), pp. 499–508.

[22] Garey, M.R., Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman and Company, San Francisco, 1979.

[23] Hochbaum, D.S., Shmoys, D.B., "Using Dual Approximation Algorithms for Scheduling Problems: Theoretical and Practical Results", *Journal of the ACM* 34:1 (January 1987), pp. 144–162.

[24] Huang, X., *Parameterized Complexity and Polynomial-Time Approximation Schemes*, PhD Thesis, Texas A&M University, 2004.

[25] Hunt III, D.B., Marathe, M.V., Radhakrishnan, V., Ravi, S.S., Rosenkrantz, D.J., Stearns, R.E., "NC-Approximation Schemes for NP- and PSPACE-Hard Problems for Geometric Graphs", *J. Algorithms* 26:2 (February 1998), pp. 238–274.

[26] Jansen, K, Zhang, G., "Maximizing the Number of Packed Rectangles" in: Hagerup, T., Katajainen, J. (eds.), *Algorithm Theory – SWAT 2004, Proc. 9th Scandinavian Workshop*, Lecture Notes in Computer Science, Volume 3111, Springer-Verlag, Berlin, 2004, pp. 362–371.

[27] Kann, V., *On the Approximability of NP-complete Optimization Problems*, Dissertation, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden, 1992.

[28] Karmarkar, N., Karp, R.M., "An Efficient Approximation Scheme for the One-Dimensional Bin Packing Problem" in: *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 1982)*, IEEE Computer Society, 1982, pp. 312–320.

[29] Khanna, S., Motwani, R., Sudan, M., Vazirani, U.V., "On Syntactic versus Computational Views of Approximability", *SIAM Journal on Computing* 28:1 (1998), pp. 164–191.

[30] Kolaitis, P.G, Thakur, M.N., "Logical Definability of NP Optimization Problems", *Information and Computation* 115:2 (December 1994), pp. 321–353.

[31] Kolaitis, P.G., Thakur, M.N., "Approximation Properties of NP Minimization Classes", *Journal of Computer and System Sciences* 50:3 (June 1995), pp. 391–411.

[32] Korte, B., Schrader, R., "On the Existence of Fast Approximation Schemes" in: Mangasarian, O.L., Meyer, R.R., Robinson, S.M. (eds.), *Nonlinear Programming 4*, Academic Press, New York, 1981, pp. 415–438.

[33] Kratsch, S., "Polynomial Kernelizations for MIN $F^+\Pi_1$ and MAX NP" in: Albers, S., Marion, J.-Y. (eds.), *26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009)*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009, pp. 601–612.

[34] Marx, D., "Parameterized Complexity of Independence and Domination on Geometric Graphs" in: Bodlaender, H.L., Langston, M.A. (eds.), *Parameterized and Exact Computation – IWPEC 2006, Proceedings of the Second International Workshop*, Lecture Notes in Computer Science, Volume 4169, Springer-Verlag, Berlin, 2006, pp. 154–165.

[35] Marx, D., "On the Optimality of Planar and Geometric Approximation Schemes" in: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, IEEE Computer Society, 2007, pp. 338–348.

[36] Orponen, P., Mannila, H., *On Approximation Preserving Reductions: Complete Problems and Robust Measures*, Technical Report C-1987-28, Department of Computer Science, University of Helsinki, 1987.

[37] Papadimitriou, C., Yannakakis, M., "Optimization, Approximation and Complexity Classes", *Journal of Computer and Systems Sciences* 43:3 (December 1991), pp. 425–440.

[38] Paz, A., Moran, S., "Non Deterministic Polynomial Optimization Problems and their Approximations", *Theoretical Computer Science* 15:3 (1981), pp. 251–277.

[39] Petrank, E., "The Hardness of Approximation: Gap Location", *Computational Complexity* 4:2 (June 1994), pp. 133–157.

[40] Queyranne, M., "Bounds for Assembly Line Balancing Heuristics", *Operations Research* 33:6 (November 1985), pp. 1353–1359.

[41] Simon, H.U., "On Approximate Solutions for Combinatorial Optimization Problems", *SIAM Journal on Discrete Mathematics* 3:2 (May 1990), pp. 294–310.

[42] van Leeuwen, E.J., "Approximation Algorithms for Unit Disk Graphs" in: Kratsch, D. (ed.), *Graph-Theoretic Concepts in Computer Science - WG 2005, Proceedings of the 31st International Workshop*, Lecture Notes in Computer Science, Volume 3787, Springer-Verlag, Berlin, 2005, pp. 351–361.

[43] van Leeuwen, E.J., "Better Approximation Schemes for Disk Graphs" in: Arge, L., Freivalds, R. (eds.), *Algorithm Theory - SWAT 2006, Proceedings of the 10th Scandinavian Workshop*, Lecture Notes in Computer Science, Volume 4059, Springer-Verlag, Berlin, 2006, pp. 316–327.

[44] van Leeuwen, E.J., *Optimization and Approximation on Systems of Geometric Objects*, PhD Thesis, University of Amsterdam, 2009.

[45] Vizing, V.G., "Ob otsenke khromaticheskogo klassa p-grafa" (Russian; On an estimate of the chromatic class of a p-graph)", *Diskretnyĭ Analiz* 3 (1964), pp. 25–30.

[46] Woeginger, G., "There is no Asymptotic PTAS for Two-Dimensional Vector Packing", *Information Processing Letters* 64:6 (December 1997), pp. 293–297.