COMPUTER SCIENCE REVIEW 4 (2010) 19–40

available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cosrev

Survey

A survey on the structure of approximation classes

*Bruno Escoffier**, *Vangelis Th. Paschos*

LAMSADE, CNRS and Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France

ARTICLE INFO

Article history:
Received 15 December 2008
Received in revised form
13 November 2009
Accepted 13 November 2009

ABSTRACT

The structure of approximability classes by the introduction of approximation preserving reductions has been one of the main research programmes in theoretical computer science during the last thirty years. This paper surveys the main results achieved in this domain.

1. Introduction

The issue of polynomial approximation theory is the construction of algorithms for **NP**-hard problems that compute "good" (under some predefined quality criterion) feasible solutions for them in polynomial time. This issue appears very early (in the seminal paper by Johnson [1]), only three years after the celebrated Cook's theorem [2]. For over thirty years, polynomial approximation has been a major research program in theoretical computer science that has motivated numerous studies by researchers from all over the world.

One of the main goals of this program is, in a first time, to provide a structure for the principal class of optimization problems, called **NPO** in what follows, and formally defined in Section 2. Providing such a structure for **NPO**, consists first of dividing it into sub-classes, called *approximability classes*, the problems belonging to each of them sharing common approximability properties (e.g., they are approximable within constant approximation ratios). Indeed, even if **NP**-hard problems can be considered equivalent regarding their optimal resolution by polynomial time algorithms, they behave very differently regarding their approximability. For instance, some **NP**-hard problems are "well" approximable (e.g., within constant approximation ratios, or within ratios arbitrarily close to 1) while there exist other problems for which such approximability qualities are impossible unless a highly improbable complexity hypothesis is true (for example **P** = **NP**).

In a second time, the objective is to structure every such sub-class by exhibiting "hard" problems for each of them. This requires a processing of the same type as for **NP**-completeness but adapted to the combinatorial characteristics of each of the approximability classes handled. This can be done by the introduction of notions of *approximability preserving reductions*. Sections 4–8 are dedicated to such reductions.

The technique of transforming a problem into another in such a way that the solution of the latter entails, somehow, the solution of the former, is a classical mathematical technique that has found wide application in computer science since the seminal works of Cook [2] and Karp [3] who introduced particular kinds of transformations (called *reductions*) with the aim of studying the computational complexity of combinatorial decision problems. The interesting aspect of a reduction between two problems consists of its twofold application: on one side it allows to transfer positive results (resolution techniques) from one problem to the other one and, on the other side, it may also be used for deriving negative (hardness) results. In fact, as a consequence of such seminal work, by making use of a specific kind of reduction, the polynomial-time *Karp-reducibility*, it has been possible to establish a complexity partial order among decision problems, which, for example, allows us to state that, modulo polynomial time transformations, the SATISFIABILITY problem is

* *Corresponding author.*
E-mail addresses: escoffier@lamsade.dauphine.fr (B. Escoffier), paschos@lamsade.dauphine.fr (V.T. Paschos).

1574-0137/$ - see front matter
doi:10.1016/j.cosrev.2009.11.001

as hard as thousands of other combinatorial decision problems, even though the precise complexity level of all these problems is still unknown. In the same spirit, approximation preserving reducibilities allow us to establish a preorder among combinatorial optimization problems with respect to their common approximability properties and independently of the particular approximability properties of each of them.

Strictly associated with the notion of reducibility is the notion of completeness. Problems that are complete in a complexity class via a given reducibility are, in a sense, the hardest problems of such a class. Besides, given two complexity classes **C** and $\mathbf{C}' \subseteq \mathbf{C}$, if a problem Π is complete in **C** via reductions that preserve membership to $\mathbf{C}'$, in order to establish whether $\mathbf{C}' \subset \mathbf{C}$ it is "enough" to assess the actual complexity of Π (informally we say that Π is a candidate to separate **C** and $\mathbf{C}'$).

The aim of this survey is to present and discuss the last developments of the structural aspects of the polynomial time approximation theory (as for example completeness in classes like **Poly-APX**, **Log-APX**, **DPTAS**). Several previous surveys and books (see, for instance, [4–8]) partially deal with this issue and contain some of the concepts and results presented here. This survey includes several recent results about completeness in approximability classes and carries out a large discussion about completeness in differential approximability classes. Besides, it is organized in such a way to emphasize the several techniques (Cook's like proofs, **PCP** theorem) that have been used to derive completeness results. Finally, as such results have been obtained for each of the notorious approximation classes, we hope that the survey provides a global structured picture of this research programme.

In what follows, we assume that the reader is familiar with basic notions from complexity theory and polynomial time approximation theory although, for sake of completeness, we provide basic definitions about complexity and approximability classes as well as about Karp and Turing reducibilities. Also, definitions of the problems discussed are omitted; they can be found in [9] or in [4]. The paper is organized as follows. The first two sections have an introductory character. In Section 2, we summarize the definitions of the basic complexity classes (**NP**, **NPO**) and we recall the notions of two important reducibilities: Karp and Turing reducibilities. In Section 3, basic notions about polynomial time approximation are presented together with the definition of the main approximation classes of optimization problems. Section 4 deals with the notion of an approximation preserving reduction and the notion of completeness in an approximation class. In Sections 5–7 the basic techniques for deriving completeness

results in approximation classes are discussed. Finally, in Section 8, completeness results for differential-approximation classes are presented.

2. Preliminaries: The classes P, NP, NPO; Karp and Turing reducibilities

In this section we revisit some basic notions of complexity theory that will be used later. More details about them can be found in [9–12].

A *decision problem* is defined by a set D of instances and a question Q. Given an instance $I \in D$, one wishes to determine if the answer to Q is *yes*, or *no*, on I. More formally, a decision problem is a set D of instances partitioned into two sets D^+ and D^-, where D^+ is the set of *positive* instances (the ones where the answer to Q is *yes*) and D^- is the set of *negative* instances, where the answer to Q is *no*. The solution of such a problem consists of determining, given $I \in D$, if $I \in D^+$ or not ($I \in D^-$).

A decision problem belongs to **NP** if there exists a polynomial binary relation R and a polynomial p such that for any instance $I \in D$, $I \in D^+$ if and only if there exists x, $|x| \leqslant p(|I|)$, such that $R(I, x)$ (in other words, given I and x, one can answer in time polynomial with $|I| + |x|$ if $R(I, x)$ or not). This amounts to saying that the class **NP** is exactly the set of problems solvable in polynomial time by a *non-deterministic* Turing machine [13,10].

Let $D_1 = (D_1^+, D_1^-)$ and $D_2 = (D_2^+, D_2^-)$ be two decision problems. Then D_1 reduces to D_2 under *Karp-reducibility* (denoted by $D_1 \leqslant_K D_2$) if there exists a function f such that:

- $\forall I \in D_1, f(I) \in D_2$;
- $I \in D_1^+ \Leftrightarrow f(I) \in D_2^+$;
- f is computable in polynomial time.

The fundamental property of this reducibility is that, *if* $D_2 \in \mathbf{P}$ and if $D_1 \leqslant_K D_2$, then $D_1 \in \mathbf{P}$.

A *problem D is said to be* **NP**-*complete if any problem of* **NP** *reduces to D under Karp-reducibility*. An immediate corollary of this definition is that *an* **NP**-*complete problem D is polynomial if and only if* $\mathbf{P} = \mathbf{NP}$. This stipulates that if $\mathbf{P} \neq \mathbf{NP}$, classes **P** and **NP**-complete are disjoint. Moreover, as it is proved in [14], these two classes do not partition **NP**. In other words, *there exist problems that are neither in* **P**, *nor* **NP**-*complete*. These are the so-called **NP**-*intermediate* problems (with respect to **P** and under Karp-reducibility). Indeed, Ladner proves in [14] that there exists an infinity of complexity-levels in the complexity-hierarchy built by the Karp-reducibility.

Let D and D' be two decision problems. Then, D reduces to D' under *Turing-reducibility* (denoted by $D \leqslant_T D'$) if, given a polynomial oracle $\bigcirc$ that solves D', there exists a polynomial time algorithm A solving D by running (calling) $\bigcirc$. A polynomial oracle for D' is a kind of fictive algorithm that solves D', i.e., it correctly determines if an instance $I \in D'^+$. So, A would remain polynomial even if it called $\bigcirc$ on a polynomial number of instances of D' derived from a single instance of D. As for Karp-reducibility, given two decision problems D and D', *if* $D' \in \mathbf{P}$ *and* $D \leqslant_T D'$, *then* $D \in \mathbf{P}$. In other words, *Turing-reducibility also preserves membership in* **P**. Hence, in a similar way as for Karp-reducibility, one can define **NP**-completeness under Turing-reducibility.

A notion of **NP**-intermediate problems under Turing-reducibility (and relatively to **P**) can also be considered. However, this reducibility still remains insufficiently surrounded. It seems to be larger than that of Karp [15], but we do not know yet if it is indeed more powerful than it, or not. The question of the existence of intermediate problems under this reducibility remains, to our knowledge, open.

We now define the class of optimization problems that have decision counterparts in **NP**. This problem-class is called **NPO**.

Definition 1. An **NP** optimization (NPO) problem Π is commonly defined (see, for example, [4]) as a four-tuple $(\mathcal{I}, \text{Sol}, m, \text{goal})$ such that:

- $\mathcal{I}$ is the set of instances of Π and it can be recognized in polynomial time;
- given $I \in \mathcal{I}$, Sol(I) denotes the set of feasible solutions of I; for every $x \in \text{Sol}(I)$, $|x|$ is polynomial in $|I|$; given any I and any x polynomial in $|I|$, one can decide in polynomial time if $x \in \text{Sol}(I)$;
- at least one $x \in \text{Sol}(I)$ can be computed in time polynomial in $|I|$;
- given $I \in \mathcal{I}$ and $x \in \text{Sol}(I)$, $m(I, x)$ is polynomial time computable and denotes the value of x for I; this value is sometimes called the "objective value";
- goal $\in \{\max, \min\}$.

Solving Π on I consists of determining a solution x^* that optimizes $m(I, x)$, $x \in \text{Sol}(I)$ (in the sense of the goal(Π). The quantity $m(I, x^*)$ will be denoted by opt(I).

From any **NPO** problem $\Pi = (\mathcal{I}, \text{Sol}, m, \text{goal})$, one can derive its decision counterpart by considering an instance $I \in \mathcal{I}$ and an integer K and by asking whether there exists a solution of value at least (resp., at most) K if goal = max (resp., min), or not. It can be easily shown that such a decision problem is in **NP**.

Given a class of problems $\mathbf{C} \subseteq \mathbf{NPO}$, we denote by **Max-C** and **Min-C** the restrictions of **C** to maximization and minimization problems, respectively.

For a combinatorial problem $\Pi \in \mathbf{NPO}$, Π is said to be *polynomially bounded* if there exists a polynomial p such that, for every instance I of Π and every $x \in \text{Sol}(I)$, $m(I, x) \leqslant p(|I|)$. Also, Π is said to be *differentially polynomially bounded*, if there exists a polynomial p such that, for every instance I of Π and every pair of solutions $(x, y) \in \text{Sol}(I) \times \text{Sol}(I)$, $|m(I, x) - m(I, y)| \leqslant p(|I|)$. Given a problem-class $\mathbf{C} \subseteq \mathbf{NPO}$, we denote by **C-PB** (resp., **C-DPB**), the restriction of **C** to polynomially bounded (resp., differentially polynomially bounded) problems.

3. Polynomial time approximation

3.1. *Approximation ratios*

The issue of polynomial approximation theory can be summarized as "the art of solving an **NP**-hard problem in polynomial time by algorithms guaranteeing a certain quality of the computed solutions". This implies the definition of quality measures for the solutions achieved; these are the so-called *approximation ratios*.

Let x be a feasible solution of an instance I of some **NPO** problem Π. The *standard approximation ratio* of x on I is defined by $\rho_\Pi(I, x) = m(I, x)/\text{opt}(I)$. This ratio is in $[0, 1]$, if goal(Π) = max, and in $[1, \infty]$, if goal(Π) = min. In both cases the closer to 1 the better the solution. Let us note that, for simplicity, when it is clear by the context, subindex Π will be omitted. Details and major results dealing with the standard approximation paradigm can be found in [4,8,12].

Even if most of the approximation results have been produced using this ratio, this is not the only approximation measure used. Already, a measure more restrictive than the standard approximation ratio is the one that measures the absolute difference between $m(I, x)$ and opt(I), i.e., the measure $|m(I, x) - \text{opt}(I)|$ [9]. But this measure in many cases is too restrictive to allow positive approximation results.

However, it would be interesting to measure the quality of a feasible solution not only with respect to an optimal one. For instance, it might be interesting to locate its value in the interval (optimal value, value of the worst solution of the instance). Another requirement could be the stability of the approximation ratio with respect to elementary transformations of the value of the problem. Clearly, consider an **NPO** problem Π', where instances, feasible solutions and the goal are the same as for problem Π but, for an instance I and a solution x, $m_{\Pi'}(I, x) = \alpha m_\Pi(I, x) + \beta$, where α and β are some non-zero constants. Both Π and Π' are "equivalent" in the sense that a solution for the one immediately provides a solution for the other. Hence, one could require that an algorithm solving Π and Π' has the same or quite similar approximation ratios for both of them. This is not the case for the standard-approximation ratio. The dissymmetry MAX INDEPENDENT SET, MIN VERTEX COVER is the most known example.

Such considerations have led to the introduction by Ausiello et al. [16] of what has been called later in [17] *differential approximation ratio*. This ratio measures the quality of a solution by comparing its value not only with the optimal value but also with the worst value of an instance, denoted by $\omega(I)$. Formally, a worst solution of an instance I of an **NPO** problem Π is defined as an optimal solution of a problem having the same set of instances and feasible solutions, but its goal being the opposite of the goal of Π. The worst solution of an instance of **NPO** problem can be easy or hard to be computed, depending on the problem at hand. For instance, computing a worst solution for an instance of MIN TSP becomes computing an optimal solution for MAX TSP on the same instance; this computation is **NP**-hard. On the other hand, a worst solution for MIN VERTEX COVER is the vertex-set of the input graph; its computation is trivial.

Let x be a feasible solution of an instance I of an **NPO** problem Π. The *differential-approximation ratio* of x in I is defined by: $\delta_\Pi(I, x) = (m(I, x) - \omega(I))/(\text{opt}(I) - \omega(I))$. By convention, when $\text{opt}(I) = \omega(I)$ (every solution of I has the same value), $\delta_\Pi(I, x) = 1$. As one can see, the differential ratio takes values in $[0, 1]$, the closer to 1, the better the ratio.

Even if it has been introduced since 1977 (hence, just few years after the paper by Johnson [1]), the differential ratio has been seldom used until the beginning of the Nineties. Indeed, the papers by [18,16,19] are, to our knowledge, the most known works using this ratio until 1993 when the beginnings of an axiomatic approach for the theory of the differential polynomial approximation that have led to more systematic use of the differential ratio have been presented in [20] and completed in [17,21,22].

Each of the two approximation paradigms induces its own results that, for the same problem, can be very different from one paradigm to the other one, in particular for minimization problems. On the other hand, for maximization problems, there is a direct relation linking the two ratios, which is stated in Proposition 1. Let Π be an optimization problem and r be a function: $r : \mathcal{I} \rightarrow \mathbb{R}$. An *approximation algorithm* A for Π is an algorithm computing, for every instance I of Π, a feasible solution x for I. Algorithm A is said to be an r-approximation algorithm if, for every instance I, the approximation ratio of x is better than $r(I)$. Problem Π is said to be *r-approximable* if there exists an r-approximation algorithm for Π.

Proposition 1. *If Π is a maximization problem, then for every instance I and every solution x of I, $\delta_\Pi(I, x) \leqslant \rho_\Pi(I, x)$. Consequently, if Π is r-differentially approximable, it is also r-standard approximable.*

One of the main stakes of polynomial approximation is, given an **NPO** problem Π, the development of algorithms achieving the best possible approximation (performance) guarantees. This is a two-fold stake. In a first time it aims at devising and analyzing polynomial approximation algorithms for obtaining fine performance guarantees for them. These are the so-called *positive results*. Another question relative to the achievement of a positive result is if the particular analysis made for a particular algorithm is the finest possible for it, i.e., if there are instances where the approximation ratio proved for this algorithm is attained (this is the so-called *tightness analysis*). This concerns particular algorithms and the underlying question is if our mathematical analysis to get the particular positive result is as fine as possible. However, there exists a more global question, addressed this time not to a single algorithm but to the problem Π itself. Informally, this stake does not only consist of answering if our analysis is good or fine, but if the algorithm devised is the best possible (with respect to the approximation ratio it guarantees). In other words, "do there exist better algorithms for Π?". Or, more generally, "what is the best approximation ratio that a polynomial time algorithm could ever guarantee for Π?". This type of results are said to be *negative*, or *inapproximability results*. Here, the challenge is to prove that Π is inapproximable within some ratio r unless a very unlikely complexity hypothesis becomes true (the strongest such hypothesis being $\mathbf{P} = \mathbf{NP}$). More details about the stake of inapproximability are given in Section 3.3.

One can now summarize the study of a problem Π in the framework of polynomial approximation. Such a study consists of:

1. establishing that Π is approximable within approximation ratio r (for the best possible r);
2. establishing that, under some complexity hypothesis (ideally $\mathbf{P} \neq \mathbf{NP}$), Π is inapproximable within some ratio r';
3. trying to minimize the gap between r and r' (the ideal being that these quantities are arbitrarily close).

Getting simultaneous satisfaction of items 1, 2 and 3 above, is not always easy; however, there exist problems for which this has been successfully achieved.[1] Let us consider, for example, MAX E3SAT. Here, we are given m clauses over a set of n boolean variables, each clause containing exactly 3 literals (a literal is a variable or its negation). The objective then is to determine an assignment of truth values to the variables that maximizes the number of satisfied clauses. In [1], it is shown that MAX E3SAT is approximable within standard approximation ratio of 7/8. On the other hand, Håstad has shown more recently in [23] that this problem is inapproximable within $7/8 + \varepsilon$, for every $\varepsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$.

3.2. *Approximation classes*

NP-hard problems behave very differently with respect to their approximability. For instance, there exist problems approximable within constant approximation ratios (e.g., MAX E3SAT, for the standard paradigm) while, for others (e.g., MAX INDEPENDENT SET), such approximability is impossible unless $\mathbf{P} = \mathbf{NP}$. The construction of approximation classes aims at building a hierarchy within **NPO**, any class of this hierarchy including problems that are "similarly" approximable.

The most notorious approximability classes widely studied until now are the following (note that, in fact, there exists a continuum of approximation classes):

- **Exp-APX** (resp., **Exp-DAPX** for the differential approximation): a problem $\Pi \in$ **NPO** is in **Exp-APX** (resp., **Exp-DAPX**), if there exists a positive polynomial p such that Π is $O(2^{p(n)})$-approximable if goal(Π) = min, or $\Omega(2^{-p(n)})$-approximable if goal(Π) = max (resp., $\Omega(2^{-p(n)})$-approximable), where n is the size of an instance of Π;
- **Poly-APX** (resp., **Poly-DAPX**): a problem $\Pi \in$ **NPO** is in **Poly-APX** (resp., **Poly-DAPX**), if there exists a positive polynomial p such that Π is $O(p(n))$-approximable if goal(Π) = min, or $\Omega(1/p(n))$-approximable if goal(Π) = max (resp., $\Omega(1/p(n))$-approximable);
- **Log-APX** (resp., **Log-DAPX**): a problem $\Pi \in$ **NPO** is in **Log-APX** (resp., **Log-DAPX**), if Π is $O(\log n)$-approximable if goal(Π) = min, or $\Omega(1/\log n)$-approximable if goal(Π) = max (resp., $\Omega(1/\log n)$-approximable);
- **APX** (resp., **DAPX**): a problem $\Pi \in$ **NPO** is in **APX** (resp., **DAPX**), if there exists a fixed constant $r > 1$ such that Π is r-approximable if goal(Π) = min, or r^{-1}-approximable if goal(Π) = max (resp., r^{-1}-approximable);
- **PTAS** (resp., **DPTAS**): a problem $\Pi \in$ **NPO** is in **PTAS** (resp., **DPTAS**) if, for any constant $\varepsilon > 0$, Π is approximable within ratio $1 + \varepsilon$ if goal(Π) = min, or $1 - \varepsilon$ if goal(Π) = max (resp., $1 - \varepsilon$); a family of algorithms $(\mathtt{A}_\varepsilon)_{\varepsilon>0}$ that guarantees such ratios (depending on the goal, or on the paradigm) is called *polynomial time (standard-, or differential-) approximation schema*;
- **FPTAS** (resp., **DFPTAS**): a problem $\Pi \in$ **NPO** is in **FPTAS** (resp., **DFPTAS**) if it admits a standard- (resp., differential-) approximation schema $(\mathtt{A}_\varepsilon)_{\varepsilon>0}$ that is polynomial in both n and $1/\varepsilon$; such a schema is called *fully polynomial time (standard-, or differential-) approximation schema*.

Finally, the class of the **NPO** problems that have decision counterparts in **P** will be denoted by **PO**. In other words, **PO** is the class of polynomial time solvable optimization problems.

Dealing with the classes defined above, the following inclusions hold:

$$\mathbf{PO} \subset \mathbf{FPTAS} \subset \mathbf{PTAS} \subset \mathbf{APX} \subset \mathbf{Log\text{-}APX} \subset \mathbf{Poly\text{-}APX} \subset \mathbf{Exp\text{-}APX} \subset \mathbf{NPO}$$

[1] Note that simultaneous satisfaction of these three items designs, in some sense, is what can ideally be done in polynomial time for the problem at hand.

These inclusions are strict unless **P** = **NP**. Indeed, for any of these classes, there exist natural problems that belong to each of them but not to the immediately smaller one. For instance:

KNAPSACK $\in$ **FPTAS** \ **PO**

MAX PLANAR INDEPENDENT SET $\in$ **PTAS** \ **FPTAS**

MIN VERTEX COVER $\in$ **APX** \ **PTAS**

MIN SET COVER $\in$ ***Log − APX*** \ **APX**

MAX INDEPENDENT SET $\in$ ***Poly − APX*** \ ***Log − APX***

MIN TSP $\in$ ***Exp − APX*** \ ***Poly − APX***

The landscape is completely similar (with respect to the architecture of the hierarchy and the strictness of the class-inclusions) for the differential paradigm modulo the fact that no natural problems are known for ***Exp − DAPX*** \ ***Poly − DAPX*** and for ***Log − DAPX*** \ **DAPX**.

Also, let us mention that in the differential hierarchy, another class can be defined, namely, the class **0-DAPX** [24]. This is the class of problems for which any polynomial time algorithm returns a worst solution on at least one instance, or, on an infinity of instances. In other words, for the problems in **0-DAPX**, their differential approximation ratio is equal to 0. More formally, a problem $\Pi \in$ **NPO** is in **0-DAPX**, if Π is not δ-differential-approximable, for any function $\delta : \mathbb{N} \to \mathbb{R}^+_*$. The first problem shown to be in **0-DAPX** is MIN INDEPENDENT DOMINATING SET, where, given a graph, one wishes to determine a minimum cardinality maximal (for the inclusion) independent set.

3.3. *Inapproximability*

We have already mentioned that the study of approximability properties of a problem includes two complementary issues: the development of approximation algorithms guaranteeing “good” approximation ratios and the achievement of inapproximability results. In this section, we focus ourselves on the second issue. We first introduce the notion of GAP-reduction that has led to the achievement of the first inapproximability results, and then, the more general technique of *approximability preserving reductions*. Next, we briefly describe the very powerful tool of the probabilistically checkable proofs upon which the major inapproximability results are based. The interested reader can also refer to [25] for further details on the theory of inapproximability.

3.3.1. *Initial technique: The GAP-reduction*

An inapproximability result for an **NPO** problem Π consists of showing that if we had an approximation algorithm achieving some approximation ratio r, then this fact would contradict a commonly accepted complexity hypothesis (e.g., **P** $\neq$ **NP**). The first idea that comes in mind is to properly adapt Karp-reducibility to this goal.

Example 1. Revisit the **NP**-completeness proof for COLORING (the decision version of MIN COLORING), given in [9]. The reduction proposed there constructs, starting from an instance φ of E3SAT, a graph G such that G is 3-colorable if φ is satisfiable, otherwise G is colorable with at least 4-colors.

Suppose now that there exists a polynomial time algorithm for MIN COLORING guaranteeing standard-approximation ratio $(4/3) - \epsilon$, with $\epsilon > 0$. Run it on the graph G constructed from φ. If φ is not satisfiable, then this algorithm computes a coloring for G using more than 4 colors. On the other hand, if φ is satisfiable (hence G is 3-colorable), then the algorithm produces a coloring using at most $3((4/3) - \epsilon) < 4$ colors, i.e., a 3-coloring. So, on the hypothesis that a polynomial time algorithm for MIN COLORING guaranteeing approximation ratio $(4/3) - \epsilon$ exists, one can in polynomial time decide if a CNF ϕ, instance of E3SAT, is satisfiable or not, contradicting so the **NP**-completeness of this problem. Hence, MIN COLORING is not $((4/3) - \epsilon)$-standard-approximable, unless **P** = **NP**.[2] ■

One can find an analogous reduction for MIN TSP in [27] (see also [9]).

The reduction of Example 1 is a typical example of a GAP-reduction. Via such a reduction one tries to create a gap separating positive instances of a decision problem from the negative ones.

More generally, starting from an instance I of an **NP**-complete decision problem D, a GAP-reduction consists of determining a polynomial time computable function f, transforming I into an instance of an **NPO** problem Π (suppose that $\text{goal}(\Pi) = \min$) such that:

- if $I \in D^+$, then $\text{opt}(f(I)) \leqslant L$;
- if $I \in D^-$, then $\text{opt}(f(I)) > U$ (for $L < U$).

Obviously, in an analogous way, one can define a GAP-reduction for the maximization problems.

Using such a reduction, one can immediately deduce an inapproximability result: Π is inapproximable within standard-ratio U/L; if not, then D would be polynomial, implying so **P** = **NP**.

3.3.2. *Generalization: approximation preserving reductions*

As we have seen in the previous section, a GAP-reduction reduces an **NP**-complete decision problem to an **NPO** problem creating a gap that can be exploited to derive inapproximability results. In what follows, we will see how one can generalize its principle in order to reduce an optimization problem Π_1 to another optimization problem Π_2. If such a reduction is carefully devised, one is able, by analogous arguments, on the hypothesis of the existence of an inapproximability result for Π_1, to get an inapproximability result for Π_2.

Example 2 ([4,8]). Let φ be an instance of MAX E3SAT on n variables and m clauses. We construct a graph $G = f(\varphi)$ instance of MAX INDEPENDENT SET as follows: for each clause of φ we build a triangle, each of its vertices corresponding to one literal of the clause. For every pair of opposite literals (e.g., x_i and $\bar{x}_i$), we add an edge in the graph under construction.

Assume now an assignment of truth values to the variables of φ satisfying k clauses. Consider in G, for each triangle corresponding to a satisfied clause, a vertex corresponding to a literal assigned with value “true”. The so-chosen set of vertices is an independent set of G of size k. Conversely, consider an independent set of size k in G, assign with “true” the literals corresponding to its vertices and arbitrarily complete the

[2] Indeed, for MIN COLORING, much stronger inapproximability results hold (see, for example, [26]).

assignment giving truth values to all the variables of φ. Obviously, this assignment satisfies at least k clauses of φ.

Putting things together, we get that an independent set S of G satisfying $m(G, S) \geqslant \text{ropt}(G)$, derives, in polynomial time, a truth assignment τ for φ satisfying $m(\varphi, \tau) \geqslant m(G, S) \geqslant \text{ropt}(G) \geqslant \text{ropt}(\varphi)$. In other words, a polynomial time algorithm achieving a standard-approximation ratio r for MAX INDEPENDENT SET, leads to a polynomial time algorithm achieving the same approximation ratio for MAX E3SAT. As a corollary, the inapproximability bound of $(7/8) + \epsilon$, for any $\epsilon > 0$, of [23] for MAX E3SAT applies also to MAX INDEPENDENT SET. So, MAX INDEPENDENT SET is inapproximable within $(7/8) + \epsilon$, for any $\epsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$.[3] ■

Approximation preserving reductions, as that of Example 2, allow the transfer of approximation results from a problem to another. This type of reduction is one of the main topics of this survey and will be addressed in Section 4. Also, additional details and informations about this type of reduction can be found in [5,6].

3.3.3. *Inapproximability results based on probabilistically checkable proofs*

The use of GAP-reductions, or of several types of approximation preserving ones, was the main tool for achieving inapproximability results until the beginning of 90's. However, they have left open several problems which, at that time, were impossible to be tackled only via this tool, as for example, the approximability of MAX INDEPENDENT SET. For this problem, ratios of $\Omega(n^{-1})$ can be very easily achieved by a bunch of polynomial time algorithms. On the other hand, MAX INDEPENDENT SET has a kind of *self-improvement* property implying that, unless $\mathbf{P} = \mathbf{NP}$, either it is approximable by a polynomial time approximation schema, or no polynomial time algorithm for it can guarantee a constant approximation ratio [9].

The major advance in the domain of inapproximability has been performed with the achievement of a new characterization of **NP**, using *probabilistically checkable proofs*. In 1991, the notion of *transparent proof* was introduced by Babai et al. [28] and has generated the exciting concept of *probabilistically checkable proofs* or *interactive proofs*. Shortly, an interactive proof system is a kind of particular conversation between a prover (P) sending a string ℓ, and a verifier (V) accepting or rejecting it; this system recognizes a language L if, (i) for any string ℓ of L (sent by P), V always accepts it, and (ii), for any $\ell \notin L$, neither P, nor any imposter substituting P, can make V accept ℓ with probability greater than 1/3. An alternative way of seeing this type of proof is as maximization problems for which the objective is to find strategies (solutions) maximizing the probability that V accepts ℓ.

Interactive proofs have produced novel very fine characterizations of the set of **NP** languages, giving rise to the development of very sophisticated gap-techniques (drawn rather in an algebraic spirit) that lead to extremely important corollaries in the seemingly unrelated domain of polynomial approximation theory.

[3] As for MIN COLORING, for MAX INDEPENDENT SET also, much stronger inapproximability results hold (see, for example, [23]).

The relation of probabilistically checkable proofs with the class **NP** have been initially studied by Feige et al., in [29] and has been completed by Arora et al., in [30]. This new characterization of **NP** by [30], known as the **PCP** theorem, has been the breakthrough for the achievement of inapproximability results for numerous **NPO** problems. In particular it has produced (unfortunately negative) answers for famous open problems in polynomial approximation as: "is MAX 3SAT in **PTAS**?", "is MAX INDEPENDENT SET in **APX**?", etc. The **PCP** theorem has been refined in the sequel, in order to produce negative results for many other well-known **NPO** problems, as for MAX E3SAT [31], MIN SET COVER [32], MIN COLORING [33–36], etc., or to refine the existing ones.

A detailed presentation of **PCP** theorem is outside the scope of this paper. However, we will try to present in what follows its main ideas. For a more detailed presentation and proofs, the interested reader can refer to [4,8]. Also, more details about the refinement of **PCP** theorem and its applications to inapproximability can be found in [7], as well as in [37,25].

Recall that **NP** is the class of problems for which there exist polynomial size certificates constituting polynomial time deterministic proofs for the fact that an instance is positive or not. Then, one can consider substitution of this deterministic verification by a randomized process. Such a process receives as inputs the instance, the certificate and a vector of random bits. The verification process will only have access to a part of the certificate, determined via the random bits. It has to decide on the positivity of the instance by only reading this part of the certificate. Of course, we wish that this random process is mistaken as rarely as possible.

The class $\mathbf{PCP}[\mathbf{r}(\mathbf{n}), \mathbf{q}(\mathbf{n})]$ is the class of problems $D = (D^+, D^-)$ for which there exists a polynomial time random algorithm RA (certificate or proof controller) such that:

- the size of vector of random bits is at most $r(n)$;
- RA has access to at most $q(n)$ bits of the certificate;
- if $I \in D^+$, then there exists a polynomial size certificate such that RA accepts with probability 1 while, if $I \in D^-$, then for any certificate of polynomial size RA accepts it with probability less than 1/2.

The fundamental theorem by [30], relates the class $\mathbf{PCP}[\cdot, \cdot]$ with the usual complexity classes and mainly with **NP**.

Theorem 1 (**PCP** *Theorem, [30]*). $\mathbf{NP} = \mathbf{PCP}[\mathbf{O}(\log(\mathbf{n}), \mathbf{O}(\mathbf{1}))]$.

Via Theorem 1 and its corollaries to approximation theory, numerous open problems have received strong answers. For instance, the **PCP** theorem has led to proofs about the inapproximability of MAX 3SAT and MAX INDEPENDENT SET by polynomial time approximation schemata. For this latter problem, this inapproximability result, combined with the self-improvement property mentioned above, allows us to conclude that MAX INDEPENDENT SET does not belong to **APX**. Note finally that the **PCP** theorem also leads, more generally, to precise inapproximability bounds for problems in **APX**. For example, as we have already mentioned, using **PCP** theorem an inapproximability bound of $7/8 - \epsilon$, for every $\epsilon > 0$, can be derived for MAX E3SAT [23].

3.3.4. *GAP-reduction vs. probabilistically checkable proofs*

As mentioned above, the **PCP** theorem is a major breakthrough in polynomial time approximation theory since it immediately entails a GAP-reduction for some optimization problems showing that these problems are hard to approximate. Really, the link between probabilistically checkable proofs and inapproximability is even stronger: the **PCP** theorem would directly follow from the existence of a particular GAP-reduction for some particular constraint satisfaction problem (CSP). More precisely, let us define CSP as follows: we are given a set of variables on a (fixed) finite domain Σ and a set $\mathcal{C}$ of constraints or arity q (q is assumed fixed) over these variables. The goal is to find an assignment to each variable such that the number of satisfied constraints is maximized. Let us denote by $\mathrm{opt}(\mathcal{C})$ the optimal value of CSP. Then, the **PCP** theorem is equivalent to the following theorem.

Theorem 2 ([38]). *There are integers $q > 1$ and $|\Sigma| > 1$ such that, given as input a collection $\mathcal{C}$ of q-ary constraints over Σ, it is* **NP**-*hard to decide whether* $\mathrm{opt}(\mathcal{C}) = |\mathcal{C}|$ *or* $\mathrm{opt}(\mathcal{C}) \leqslant |\mathcal{C}|/2$.

A challenging issue was then to prove directly the previous theorem (or, equivalently, the **PCP** theorem) using purely combinatorial arguments (the original proof of the **PCP** theorem is strongly algebraic). Such a combinatorial proof has been finally obtained recently by Dinur [38]. It is based upon the standard technique of GAP — and approximation preserving reductions, but involves a very complex and clever construction. The details of this proof are outside the scope of this survey.

Concerning the topic dealt in this survey, namely the completeness in approximation classes, the work by [38] also implies some completeness results (namely, at least the results presented in Section 7), since, as mentioned, it is equivalent to the **PCP** theorem. However, for historical reasons, and also because as mentioned in the beginning of this section, the **PCP** theorem has constituted a major breakthrough in polynomial time approximation theory, we present in this survey the proofs derived from this theorem, as they appeared in the literature.

3.4. *Logic, complexity, optimization and approximation*

Complexity theory is classically founded on language theory [13]. An alternative approach based upon a logical definition of **NP** has been proposed by Fagin in [39] and since the end of 80's has been fruitfully used in polynomial approximation theory. In this section, we briefly present this approach. It is based upon the seminal work of Papadimitriou and Yannakakis [40] exhibiting strong links between logic, complexity and approximation.

Let us first present the logical characterization of **NP**. Fagin, trying to solve an open question in mathematical logic[4] has shown that answering it was equivalent to determining if **P** = **NP**. By this way, he has provided an alternative characterization of **NP** in terms of mathematical logic. Before giving it, let us, for readability, explain it by means of an example.

Example 3. Given a graph $G(V, E)$, 3-COLORING consists of determining if G is 3-colorable or not. This problem is obviously in **NP**. The graph G can be seen as a set $V = \{v_1, \ldots, v_n\}$ of vertices (called the universe in terms of logic), together with a set E of edges that can be represented as a binary relation on $V \times V$, $E(x, y)$ meaning that edge (x, y) exists in G. A 3-coloring in G is a partition of V into three independent sets S_1, S_2 and S_3. Then, representing a set $V' \subseteq V$ as a unary relation on V ($V'(u)$ meaning that $u \in V'$), one can characterize any 3-colorable graph as one for which the following formula is true:

$$\begin{aligned}
\exists S_1 \exists S_2 \exists S_3 \quad & \big[(\forall x S_1(x) \vee S_2(x) \vee S_3(x)) && (1)\\
& \wedge \big(\forall x\, (\neg S_1(x) \wedge \neg S_2(x)) \vee (\neg S_1(x) \wedge \neg S_3(x)\big) && (2)\\
& \quad \vee \big(\neg S_2(x) \wedge \neg S_3(x))\big) && (3)\\
& \wedge \big(\forall x \forall y\, ((S_1(x) \wedge S_1(y)) \vee (S_2(x) \wedge S_2(y)) && (4)\\
& \quad \vee (S_3(x) \wedge S_3(y)))\big)\big] && (5)\\
& \quad \Rightarrow \neg E(x, y) && (6)
\end{aligned}$$

Indeed, considering that S_1, S_2 and S_3 are the three colors, line (1) means that every vertex has to receive at least one color, and lines (2) and (3) that every vertex has to receive at most one color. Finally, lines (4) to (6) model the fact that adjacent vertices cannot receive the same color. ■

Example 3 above shows that 3-COLORING can be expressed by means of a second-order logical formula, positive instances of it being the ones satisfying it. This can be generalized for any problem in **NP**.

Let $n \in \mathbb{N}$ and $\sigma = (\sigma_1, \ldots, \sigma_n) \subset \mathbb{N}^n$. A σ-structure is an $(n+1)$-tuple $(A, P_1, \ldots, P_n)$ where: A is a non-empty finite set called the *universe* and P_i is a predicate of arity σ_i on A. Fagin's theorem claims that for any decision problem $D = (D^+, D^-)$ of **NP**, there exist $\sigma \in \mathbb{N}^{n_1}$, $\mu \in \mathbb{N}^{n_2}$, and a first-order formula Φ such that D can be modeled in the following way:

- every instance I of D can be represented by a σ-structure I_σ;
- an instance I is positive if and only if there exists a μ-structure $\mathcal{S}$ such that $(I_\sigma, \mathcal{S}) \models \Phi(I_\sigma, \mathcal{S})$.

A detailed proof of this theorem can also be found in [10].

Based upon the NP characterization provided by Fagin, the work presented in [40] consists of determining natural syntactic classes of **NPO** problems, each of them having particular approximation properties.

Consider a problem $D \in$ **NP**, expressed by formula $\exists \mathcal{S}$ $(I, \mathcal{S}) \models \Phi(I, \mathcal{S})$, Φ being a first-order formula. If it starts from $\forall$, one can define the optimization problem Π_D, "corresponding" to D, as follows:

$$\Pi_D = \max_S \left|\left\{x : (I, S) \models \Phi'(x, I, S)\right\}\right|$$

where Φ' is the formula obtained from Φ by removing the first quantifier and x is a k-tuple of elements from the universe of I.

Then, the most important classes of [40] can be defined as follows.

Definition 2 ([40]). Consider a quantifier-free first-order logic formula ϕ. Then:

- **Max-SNP** is the set of maximization problems defined by: $\max_S |\{x : (I, S) \models \phi(x, I, S)\}|$; the objective of any of these problems is to determine a structure S^* maximizing quantity $|\{x : (I, S) \models \phi(x, I, S)\}|$;

[4] Is the complement of a generalized spectrum always a generalized spectrum or not?

- **Max-NP** is the set of maximization problems defined by:

  $\max_S |\{x : (I, S) \models \exists y \phi(x, y, I, S)\}|$.

  The objective is to determine a structure S^* maximizing quantity:

  $|\{x : (I, S) \models \exists y \phi(x, y, I, S)\}|$. ■

Definition 2 does not really fit the formal framework of **NPO** problems. In this definition, structures S represent, in some sense, the feasible solutions of an instance. So, considering for an instance I, $m(I, S) = |\{x : (I, S) \models \phi(x, I, S)\}|$, for **Max-SNP**, or $m(I, S) = |\{x : (I, S) \models \exists y \phi(x, y, I, S)\}|$, for **Max-NP**, one can recover the formal framework of **NPO**.

Many well-known optimization problems belong to classes **Max-SNP** and **Max-NP**. For instance, MAX 3SAT, MAX INDEPENDENT SET-B (the restriction of MAX INDEPENDENT SET to graphs of maximum degree bounded by B), MAX CUT, etc., belong to **Max-SNP**, while MAX SAT belongs to **Max-NP**.

The following theorem [40] establishes the first link between **Max-SNP**, **Max-NP** and approximation theory.

Theorem 3 ([40]). *Every problem in **Max-SNP** and in **Max-NP** is approximable within constant standard-approximation ratio. More precisely, **Max-SNP** $\subset$ **Max-NP** $\subset$ **APX**.*

4. Reductions and completeness

When the problem of characterizing approximation algorithms for hard optimization problems was tackled, soon the need arose for a suitable notion of reducibility that could be applied to optimization problems in order to study their approximability properties.

What is it that makes algorithms for different problems behave in the same way? Is there some stronger kind of reducibility than the simple polynomial reducibility that will explain these results, or are they due to some structural similarity between the problems as we define them? [1]

Approximation preserving reductions provide an answer to the above question. Such reductions have an important role when we wish to assess the approximability properties of an **NPO** optimization problem and locate its position in the approximation hierarchy. In such case, in fact, if we can establish a relationship between the given problem and other known optimization problems, we can derive both positive information on the existence of approximation algorithms (or approximation schemes) for the new problem or, on the other side, negative information, showing intrinsic limitations to approximability. With respect to reductions between decision problems, reductions between optimization problems have to be more elaborate. Such reductions, in fact, have to map both instances and solutions of the two problems, and they have to preserve, so to say, the optimization structure of the two problems.

The first examples of reducibility among optimization problems were introduced by Ausiello, d'Atri and Protasi in [16,41] and by Paz and Moran in [42]. In particular in [41] the notion of *structure preserving reducibility* is introduced and for the first time the completeness of MAX WSAT (this problem will be defined later in Section 5.1) in the class of **NPO** problems is proved. Still it took a few more years until suitable notions of approximation preserving reducibilities were introduced by Orponen and Mannila in [43]. In particular their paper presented the *strict reduction* (see Section 5.1) and provided the first examples of natural problems which are complete under approximation preserving reductions.

4.1. *Approximation preserving reductions*

Let us revisit Example 2 of Section 3.3.2. The reduction from MAX E3SAT to MAX INDEPENDENT SET given in this example consists of transforming:

- an instance φ of MAX E3SAT into an instance $G = f(\varphi)$ of MAX INDEPENDENT SET;
- a feasible solution S of G into a feasible solution $g(\varphi, S) = \tau$ of φ, so that standard-approximation ratios of solutions S and τ are related by: $\rho(G, S) \leqslant \rho(\varphi, \tau)$.

This reduction allows one to claim that, if MAX INDEPENDENT SET is approximable within standard-approximation ratio r, for some constant r, then MAX E3SAT is so. Conversely, it also allows one to claim that, if MAX E3SAT cannot be approximated within standard-approximation ratio r' (under some complexity hypothesis), so does MAX INDEPENDENT SET (under the same hypothesis).

This example summarizes the principle and the interest of the notion of an approximation preserving reduction. This kind of reduction comprises three basic components:

1. a polynomial time computable function f that transforms instances of some (initial) problem Π into instances of some other (final) problem Π';
2. a polynomial time computable function g that, for every instance I of Π and every feasible solution x' of $f(I)$, returns a feasible solution $x = g(I, x')$ of I;
3. a property concerning transfer of approximation ratios from x' to x; this property implies that if x' attains some approximation level, then $x = g(I, x')$ also attains some approximation level (possibly different from the one of x').

If such a reduction exists from Π to Π' (both being **NPO** problems), one can use an approximation algorithm $\mathtt{A}'$ for Π' in order to solve Π in the following way: one transforms an instance I of Π into an instance $f(I)$ of Π', one runs $\mathtt{A}'$ in $f(I)$ and finally uses g in order to finally recover a feasible solution for I. This specifies an approximation algorithm $\mathtt{A}$ for Π which, for an instance I returns solution $\mathtt{A}(I) = g(I, \mathtt{A}'(f(I)))$. Obviously, $\mathtt{A}$ runs in polynomial time if so does $\mathtt{A}'$.

In order to further clarify this brief introduction to approximation preserving reductions we give two examples of very frequently used reductions: the L-reducibility (Definition 3) and the AF-reducibility (Definition 4).

Definition 3 ([40]). A problem $\Pi \in$ **NPO** reduces to a problem $\Pi' \in$ **NPO** under the L-reducibility (denoted by $\Pi \leqslant_{\mathrm{L}} \Pi'$) if there exist two functions f et g computable in polynomial time and two constants α et β such that:

1. for every $I \in \mathcal{I}_{\Pi}$, $f(I) \in \mathcal{I}_{\Pi'}$; furthermore, $\mathrm{opt}_{\Pi'}(f(I)) \leqslant \alpha \mathrm{opt}_{\Pi}(I)$;

2. for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $g(I, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; furthermore:

$$\left|m_{\Pi}(I, g(I, x)) - \mathrm{opt}_{\Pi}(I)\right| \leqslant \beta \left|m_{\Pi'}(f(I), x) - \mathrm{opt}_{\Pi'}(f(I))\right|. \quad \blacksquare$$

Let us note that conditions 1 and 2 imply that $|\rho_{\Pi}(I, g(I, x)) - 1| \leqslant \alpha\beta|\rho_{\Pi'}(f(I), x) - 1|$. Hence, if, for example, Π and Π' are both maximization problems and if x is a ρ'-approximated solution for $f(I)$, then $\rho_{\Pi}(I, g(I, x)) \geqslant 1 - \beta\alpha(1 - \rho')$. This allows us to get the following fundamental property of L-reducibility (holding also for minimization problems): *if* $\Pi' \in$ **PTAS** *and if* $\Pi \leqslant_{\mathrm{L}} \Pi'$, *then* $\Pi \in$ **PTAS**; in other words, *L-reducibility preserves membership in* **PTAS**. Let us note that under suitable hypotheses, the L-reducibility can also preserve membership in **APX**.

Example 4. Let us show that MAX INDEPENDENT SET-B L-reduces to MAX 2SAT. Consider an instance $G(V, E)$ of MAX INDEPENDENT SET-B. We build the following instance φ of MAX 2SAT:

- with any vertex $v_i \in V$, we associate a variable x_i;
- for every edge $(v_i, v_j) \in E$ we build the clause $\bar{x}_i \vee \bar{x}_j$ and for every vertex v_i we build the clause x_i.

This specifies function f of the reduction.

We now specify function g. Consider an assignment τ of truth values to the variables of φ. Transform τ into another assignment τ' using the following rule: if a clause $\bar{x}_i \vee \bar{x}_j$ is not satisfied, change the value of x_i (i.e., set it to "false"). The so-obtained assignment τ' satisfies at least as many clauses as τ. Consider now the set V' of vertices corresponding to variables that have been set to "true" by τ'. This set is an independent set of G since all the clauses corresponding to E are verified. Function g is so specified.

It remains to show that the reduction just described is an L-reduction. We first note that $m(\varphi, \tau) \leqslant m(\varphi, \tau') = |E| + |V'|$. Given that the maximum degree of G is bounded by B, $|E| \leqslant nB/2$ and, on the other hand, there exists an independent set of size at least $n/(B+1)$ [44]. Based upon these two facts, we get: $|E| \leqslant B(B+1)\mathrm{opt}(G)/2$. In this way, any independent set of size k in G can be transformed into a truth assignment τ satisfying $|E| = k$ clauses. Hence, $\mathrm{opt}(\varphi) = |E| + \mathrm{opt}(G) \leqslant ((B(B+1)/2) + 1)\mathrm{opt}(G)$. Condition 1 of Definition 3 is verified, taking $\alpha = B(B+1)/2 + 1$. On the other hand, $\mathrm{opt}(\varphi) - m(\varphi, \tau) \geqslant \mathrm{opt}(\varphi) - m(\varphi, \tau') = \mathrm{opt}(G) - m(G, V')$. This is condition 2 with $\beta = 1$. $\blacksquare$

The second reducibility, called the AF-reducibility, is relevant for the differential-approximation paradigm.

Definition 4 ([5]). A problem $\Pi \in$ **NPO** reduces to a problem $\Pi' \in$ **NPO** under the AF-reducibility (denoted by $\Pi \leqslant_{\mathrm{AF}} \Pi'$) if there exist two functions f et g, computable in polynomial time, and two constants $\alpha \neq 0$ et β such that:

- for every $I \in \mathcal{I}_{\Pi}$, $f(I) \in \mathcal{I}_{\Pi'}$;
- for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $g(I, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; furthermore, $m_{\Pi}(I, g(I, x)) = \alpha m_{\Pi'}(f(I), x) + \beta$;
- for every $I \in \mathcal{I}_{\Pi}$, function $g(I, .)$ is onto;
- Π and Π' have the same goal if $\alpha > 0$, and opposite goals if $\alpha < 0$. $\blacksquare$

Thanks to the fact that function $g(I, .)$ is onto, opt et ω verify $\mathrm{opt}_{\Pi}(I) = \alpha\mathrm{opt}_{\Pi'}(f(I)) + \beta$ and $\omega_{\Pi}(I) = \alpha\omega_{\Pi'}(f(I)) + \beta$. Hence, *if* $\Pi' \in$ **DPTAS** (*resp.*, $\Pi' \in$ **DAPX**) *and if* $\Pi \leqslant_{\mathrm{AF}} \Pi'$, *then* $\Pi \in$ **DPTAS** (*resp.*, $\Pi \in$ **DAPX**). In other words, *AF-reducibility preserves membership in both* **DAPX** *and* **DPTAS**.

4.2. *Completeness in approximability classes*

Every reducibility can be seen as a binary hardness-relation among problems. In general reducibilities are reflexive and transitive; this is true with all known reducibilities. In other words, they induce a *partial preorder*[5] on the set of problems linked by them. This preorder is partial because it is possible for two problems to be incomparable with respect to a reducibility, i.e., it is possible that there exist two problems Π and Π' such that, for some reducibility R, neither $\Pi \leqslant_{\mathrm{R}} \Pi'$, nor $\Pi' \leqslant_{\mathrm{R}} \Pi$.

Starting from a relation that is reflexive and transitive, one can define an associated equivalence relation, in a very classical manner. Let R be a reducibility. We define the equivalence relation $\equiv_{\mathrm{R}}$ by $\Pi \equiv_{\mathrm{R}} \Pi'$ if and only if $\Pi \leqslant_{\mathrm{R}} \Pi'$ and $\Pi' \leqslant_{\mathrm{R}} \Pi$. Then, Π et Π' are said to be equivalent under R-reducibility.

Given a set **C** of problems and a reducibility R, it is natural to ask if there exist maximal elements in the preorder induced by R, i.e., if there exist problems $\Pi \in$ **C** such that any problem $\Pi' \in$ **C**, R-reduces to Π. Such maximal elements are called in complexity theory *complete problems*.

Let **C** be a class of problems and R be a reducibility. A problem $\Pi \in$ **C** is said to be **C**-complete (under R-reducibility) if for any $\Pi' \in$ **C**, $\Pi' \leqslant_{\mathrm{R}} \Pi$. A **C**-complete problem (under reducibility R) is then (in the sense of this reducibility) a computationally hardest problem for class **C**. For instance, in the case of **NP**-completeness, **NP**-complete problems (under Karp-reducibility) are the hardest problems of **NP** since if one could polynomially solve just one of them, then one would be able to solve in polynomial time any other problem in **NP**. Let **C** be a class of problems and R a reducibility. A problem Π is said to be **C**-hard (under R-reducibility) if for any $\Pi' \in$ **C**, $\Pi' \leqslant_{\mathrm{R}} \Pi$. In other words, a problem Π is **C**-complete if and only if $\Pi \in$ **C** and Π is **C**-hard.

Finally, from a structural point of view, it is interesting to determine the *closure* of a problem-class under a given reducibility, this closure being defined as the set of problems reducible under this reducibility to a problem of the class under consideration. We could see the closure of a class as a set of problems computationally "easier" than at least one of the problems of this class. Formally, *the closure of a problem-class* **C** *under some reducibility R is defined by:* $\bar{\mathbf{C}}^{\mathrm{R}} = \{\Pi : \exists \Pi' \in \mathbf{C}, \Pi \leqslant_{\mathrm{R}} \Pi'\}$.

4.3. *Links with approximation*

It hopefully has been obvious that the concept of approximation preserving reductions is very strongly linked to polynomial approximation. This link is mainly used for the achievement of inapproximability results. Reductions preserving approximability generalize, in some sense, the concept of GAP-reduction and allow the transfer of inapproximability bounds from one optimization problem to another.

Also, another very important use of approximability preserving reductions is that they constitute the central tool

[5] We speak about preorder and not about order because reducibilities are not, in general, antisymmetric.

for completing the structure of approximability classes defined in Section 3.2 by providing complete problems for them.

Let us revisit the L-reducibility introduced in Section 4.1. We have stated there that if a problem $\Pi' \in$ **PTAS** and if a problem Π L-reduces to Π', then $\Pi \in$ **PTAS**. In other words *L-reducibility preserves membership in* **PTAS**. Consider now an approximation class that contains **PTAS**, say **APX** and assume that one has proved the existence of a problem Π that is **APX**-complete under L-reducibility. If Π admits a polynomial time approximation schema then, since L-reducibility preserves membership in **PTAS**, one can deduce the existence of polynomial time approximation schemata for any problem that is L-reducible to Π, hence, in particular, for any problem in **APX**. In other words, by the assumptions just made, we have: $\Pi \in$ **PTAS** $\Rightarrow$ **APX** $=$ **PTAS**. Since, under the hypothesis **P** $\neq$ **NP**, **PTAS** $\subsetneq$ **APX**, one can conclude that, under the same hypothesis, $\Pi \notin$ **PTAS**.

The above schema of reasoning can be generalized for any approximation class. Let **C** be a class of problems. We say that a reducibility R preserves membership in **C**, if for every pair of problems Π and Π': if $\Pi \in$ **C** and $\Pi' \leqslant_{\mathrm{R}} \Pi$, then $\Pi' \in$ **C**. We then have the following.

Proposition 2. *Let* **C** *and* **C**$'$ *be two problem-classes with* **C**$' \subsetneq$ **C**. *If a problem* Π *is* **C**-*complete under some reducibility preserving membership in* **C**$'$, *then* $\Pi \notin$ **C**$'$.

Obviously, if the strict inclusion of classes is subject to some complexity hypothesis, the conclusion $\Pi \notin$ **C**$'$ is subject to the same hypothesis.

The analogy with **NP**-completeness is immediate. The fundamental property of Karp- (or Turing-) reducibility is that it preserves membership in **P**. Application of Proposition 2 to **NP**-completeness framework simply says that **NP**-complete problems cannot be in **P**, unless **P** $=$ **NP**. Let us continue for a while this analogy. As we have mentioned in Section 2, [14] establishes the existence of **NP**-intermediate problems under Karp-reducibility. This becomes to say that there are not only two levels in the hierarchy induced by Karp-reducibility (problems in **P** and **NP**-complete problems), or, said in another way, under this reducibility and the assumption **P** $\neq$ **NP**, **NP**-completeness is not equivalent to "non-polynomiality". This type of question can be also asked when dealing with approximability-hierarchy. In other words, "can we say, for example, that a problem in **APX** that does not admit a polynomial time approximation schema is **APX**-complete?". This is the question of intermediate problems in approximation classes.

Let **C** and **C**$'$ be two problem-classes with **C**$' \subseteq$ **C**, and R be a reducibility preserving membership in **C**$'$. We say that a problem Π is **C**-*intermediate (under R-reducibility and with respect to* **C**$'$) if, under the hypothesis **C**$' \subsetneq$ **C**, Π is neither **C**-complete under R-reducibility, nor it belongs to **C**$'$.

In what follows, in Section 5, 6 et 7 we describe the three main techniques used to derive completeness results for several approximation classes.

5. Completeness by Cook's like proofs

In this section we survey completeness in standard approximation paradigm for three approximation classes, namely, **NPO** (Section 5.1), **APX** (Section 5.2) and **PTAS** (Section 5.3). Achievement of these results is based upon careful (and sometimes tricky) transformations of the proof of Cook's theorem [2] by adapting it to an "optimization" context and without using the **PCP** theorem (Theorem 1 in Section 3.3.3).

5.1. *NPO-completeness*

NPO-completeness has been initially introduced and studied by Orponen and Mannila in [43]. It is based upon a very natural idea of approximation preserving reducibility, informally, that function g returns a solution $g(\mathrm{I}, x)$ that is at least as good as x itself.

Definition 5 ([43]). Let Π and Π' be two maximization problems of **NPO**. Then Π reduces to Π' under the *strict reducibility* if there exist two functions f and g, computable in polynomial time, such that:

- for every $\mathrm{I} \in \mathcal{I}_{\Pi}$, $f(\mathrm{I}) \in \mathcal{I}_{\Pi'}$;
- for every $\mathrm{I} \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(\mathrm{I}))$, $g(\mathrm{I}, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$;
- for every $\mathrm{I} \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(\mathrm{I}))$, $\rho_{\Pi}(\mathrm{I}, g(\mathrm{I}, x)) \geqslant \rho_{\Pi'}(f(\mathrm{I}), x)$. ■

In an analogous way strict-reducibility can be defined for minimization problems.

By its simplicity, strict reducibility has the advantage to preserve membership in any approximation class and, simultaneously, the drawback to be too restrictive, since it does not allow any "deterioration" on the quality of the solution derived by g. However, this concept has allowed to derive the first completeness results achieved for approximability classes that, even if somewhat partial, have their own mathematical and historical value.

Consider problems MAX WSAT and MIN WSAT. For both of them, instances are instances of SAT, i.e., CNF's φ on m clauses and n binary variables, each variable x_i having weight $w(x_i)$. Feasible solutions for these problems are the models of φ. The value of a model τ is the sum of the weights on the variables set to "true", i.e., quantity $\sum_{i=1}^{n} w(x_i)\tau(x_i)$. The objective is to maximize (for MAX WSAT), or to minimize (for MIN WSAT) this value over any model of φ. However, so defined these two problems do not fit the definition of **NPO** (Definition 1, Section 2). Indeed, in order to find a feasible solution for φ (when such a solution exists) one has to solve SAT, impossible in polynomial time unless **P** $=$ **NP**. In order to remedy to this, and make that both problems belong to **NPO**, we add a trivial additional solution, namely, the one where all variables are set to "false" (for MAX WSAT; this solution has value 0), or the one where all variables are set to "true" (for MIN WSAT; the value of this solution is, obviously, $\sum_{i=1}^{n} w(x_i)$).

Theorem 4 ([43]). MIN WSAT *is* ***Min-NPO****-complete under strict-reducibility.*

An analogous result has been shown in [16] for **Max-NPO**.

Theorem 5 ([16]). MAX WSAT *is* ***Max-NPO****-complete under strict-reducibility.*

Let us note that in [43], the **Min-NPO**-completeness of many other problems is also shown: MIN W3SAT, MIN TSP and MIN LINEAR PROGRAMMING 0-1 (see also [45,42] for alternative proofs of these results).

Theorems 4, or 5 leave however the question of the existence of complete problems open for the whole class **NPO**. No such result has been produced using strict-reducibility. This seems to be due to a certain dissymmetry between maximization and minimization problems. More precisely, Kolaitis and Thakur show in [46] that rewriting every maximization problem into a minimization one, on the same set of instances and in such a way that optima coincide, is equivalent to showing that **NP** = **co-NP**. So, it seems necessary that, in order to prove a completeness result for (the whole class) **NPO**, one has to use a reducibility that does not preserve optimality.

This has led Crescenzi et al. to propose, in [47], a new reducibility, the AP-reducibility defined as follows.

Definition 6 ([47]). Let Π and Π' be two **NPO** problems. Then, Π reduces to Π' under AP-reducibility (denoted by $\Pi \leqslant_{\mathrm{AP}} \Pi'$), if there exist two functions f and g and a positive constant α such that:

- for every $I \in \mathcal{I}_{\Pi}$ and for every $r > 1$, $f(I, r) \in \mathcal{I}_{\Pi'}$; f is polynomial with $|I|$;
- for every $I \in \mathcal{I}_{\Pi}$, every $r > 1$ and every $x \in \mathrm{Sol}(f(I, r))$, $g(I, r, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; g is polynomial with both $|I|$ and $|x|$;
- for every $I \in \mathcal{I}_{\Pi}$, every $r > 1$ and every $x \in \mathrm{Sol}(f(I, r))$, $R_{\Pi'}(f(I, r), x) \leqslant r$ implies $R_{\Pi}(I, g(I, r, x)) \leqslant 1 + \alpha(r - 1)$, where R is approximation ratio ρ for a minimization problem and $1/\rho$ for a maximization problem. ■

Let us note that *AP-reducibility preserves membership in* **APX** *and in* **PTAS**. On the other hand, it is much less restrictive than strict-reducibility. In particular, the fact that functions f et g depend on r allows that optimal solutions are not transformed into optimal solutions and so the problem of the dissymmetry between maximization and minimization problems mentioned before can be more easily overcome.

By means of this reducibility, Crescenzi et al. prove in [47] the existence of complete problems for **NPO**. More precisely, they first prove the following theorem.

Theorem 6 ([47]). MIN WSAT *and* MAX WSAT *are equivalent under AP-reducibility.*

Let Π be, say, a **Max-NPO**-complete problem under AP-reducibility and let Π' a minimization problem of **NPO**. Then, $\Pi' \leqslant_{\mathrm{AP}}$ MIN WSAT $\leqslant_{\mathrm{AP}}$ MAX WSAT $\leqslant_{\mathrm{AP}} \Pi$, where the last reduction is an application of Theorem 4. Transitivity of AP-reducibility derives then the following corollary.

Corollary 1 ([47]). *Every problem* **Max-NPO***-complete or* **Min-NPO***-complete under AP-reducibility is* **NPO***-complete (under the same reducibility).*

Now, it suffices to remark that strict-reducibility is a particular case of AP-reducibility (taking $\alpha = 1$ and considering that f and g do not depend on r), and to use Theorems 4 and 5 and Corollary 1, in order to deduce the following corollary.

Corollary 2 ([47]). MAX WSAT, MIN WSAT, MIN W3SAT, MIN TSP *and* MIN LINEAR PROGRAMMING 0-1 *are* **NPO***-complete under AP-reducibility.*

The five problems of Corollary 2 are all weighted (variable-weights for the first three ones, edge-weights for the fourth and weights on coefficients for the fifth one) and these weights can be large. For instance, in the proof of Theorem 4 [43], weights on the variables are exponential. The fact of using such large weights is not fortuitous, as it is shown in the following theorem proved in [48].

Theorem 7 ([48]). *A polynomially bounded* **NPO** *problem cannot be* **NPO***-complete (under AP-reducibility), unless the polynomial hierarchy collapses.*

In fact, Theorem 7 is proved in [48], under PTAS-reducibility (introduced in Section 5.2) which is more general than AP-reducibility. Hence, this result still holds for this latter reducibility.

The result stated in Theorem 7 shows the difficulty of establishing a generic reduction of the whole class **NPO** to a polynomially bounded problem. So, another question arises: "do there exist complete problems for **NPO-PB**, the subclass of polynomially bounded **NPO** problems?". Consider MIN LINEAR PROGRAMMING 0-1-PB and MAX LINEAR PROGRAMMING 0-1-PB, the restrictions of MIN LINEAR PROGRAMMING 0-1 and MAX LINEAR PROGRAMMING 0-1, respectively, to the case where coefficients in the objective function are binary. Both of these problems are obviously in **NPO-PB**. As it is shown in [47], MIN LINEAR PROGRAMMING 0-1-PB *and* MAX LINEAR PROGRAMMING 0-1-PB *are* **NPO-PB***-complete under AP-reducibility*. Let us note that, as for the case of **NPO**-completeness, this result is obtained by showing that these problems are AP-equivalent (see Section 4.2) and by using former results of partial completeness by Berman and Schnitger [49], and Kann [50].

5.2. APX-completeness

We discuss in this section completeness for **APX**, the most relevant and intensively studied approximation class of **NPO**. In a first time, we present completeness results for this class under two reducibilities preserving membership in **PTAS**. Next, we present results tackling the existence of intermediate problems for **APX**. Finally, we explore the limits of the approach consisting of establishing completeness results exclusively based upon the notion of approximation preserving reductions.

5.2.1. PTAS preserving reducibilities and completeness in APX

The first completeness result for **APX** has been established by Crescenzi and Panconesi in [51]. For doing this, they introduce the P-reducibility that preserves membership in **PTAS**.

Definition 7 ([51]). Let Π and Π' be two maximization problems of **NPO**. Then, Π reduces to Π' under P-reducibility if there exist two polynomial time computable functions f et g and a function c such that:

- for every $I \in \mathcal{I}_{\Pi}$, $f(I) \in \mathcal{I}_{\Pi'}$;
- for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $g(I, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$;
- $c:]0, 1[\rightarrow]0, 1[$;
- for every $I \in \mathcal{I}_{\Pi}$, every $x \in \mathrm{Sol}(f(I))$ and every $\varepsilon \in]0, 1[$, $\rho_{\Pi'}(f(I), x) \geqslant 1 - c(\varepsilon)$ implies $\rho_{\Pi}(I, g(I, x)) \geqslant 1 - \varepsilon$. ■

P-reducibility where at least one among Π and Π' is a minimization problem can be defined analogously. Also, it can be immediately seen that *P-reducibility preserves membership in* **PTAS**.

Via this reducibility, it is proved in [51] the **APX**-completeness of a restriction of MAX WSAT, called MAX WSAT-*B*. This problem is defined as MAX WSAT modulo the fact that in every of its instances $W \leqslant \sum_{i=1}^{n} w(x_i) \leqslant 2W$, for a given integer W. As for MAX WSAT, we assume ad hoc that the assignment consisting of setting all variables to "false" is feasible and its value is W. This problem is trivially in **APX** since this ad hoc solution has standard-approximation ratio 2.

Theorem 8 (*[51]*). MAX WSAT-*B is* ***APX****-complete under P-reducibility.*

As in the case of **NPO**-completeness in Section 5.1, the proof of Theorem 8 is based upon a modification of Cook's theorem. This result has a great importance since it constitutes the first completeness result for **APX**. However, it presents two relative drawbacks: (i) MAX WSAT-*B* seems to be somewhat artificial and constructed ad hoc and (ii) it is obvious that this problem does not admit a polynomial time approximation schema, unless **P** = **NP** (in any case, any approximation better than 1/2 in polynomial time, would allow to solve SAT).

Also, it very soon appeared that it was extremely difficult to prove **APX**-completeness of other problems, under P-reducibility. Face to this situation, Crescenzi and Trevisan have shown in [52] that this reducibility is not well suited for proving completeness about polynomially bounded problems. This "inadequacy" resides into the fact that P-reducibility transforms optimum solutions into optimum solutions (as also other reducibilities do, such as the above defined strict reducibility, L-reducibility, E-reducibility).

Denote by $\mathbf{P}^{\mathrm{SAT}}$ and $\mathbf{P}^{\mathrm{SAT}[O(\log(n))]}$ the classes of decision problems solvable by using, respectively, a polynomial, or a logarithmic number of calls to an oracle solving SAT. The authors of [52] show that proving completeness in **APX** of a polynomially bounded problem under a reducibility preserving optimality of solutions implies that $\mathbf{P}^{\mathrm{SAT}} = \mathbf{P}^{\mathrm{SAT}[O(\log(n))]}$, that is again a very unlikely assumption [52]. In order to get round this difficulty, they introduce the following reducibility, called PTAS-reducibility.

Definition 8 ([52]). Let Π and Π' two maximization problems of **NPO**. Then Π reduces to Π' under PTAS-reducibility, if there exist three functions f, g and c such that:

- for every $I \in \mathcal{I}_{\Pi}$ and every $\varepsilon \in]0,1[$, $f(I,\varepsilon) \in \mathcal{I}_{\Pi'}$; f is computable in time polynomial with $|I|$;
- for every $I \in \mathcal{I}_{\Pi}$, every $\varepsilon \in]0,1[$ and every $x \in \mathrm{Sol}(f(I,\varepsilon))$, $g(I,\varepsilon,x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; g is computable in polynomial time with respect to both $|I|$ and $|x|$;
- $c :]0,1[\rightarrow]0,1[$;
- for every $I \in \mathcal{I}_{\Pi}$, every $\varepsilon \in]0,1[$ and every $x \in \mathrm{Sol}(f(I,\varepsilon))$, $\rho_{\Pi'}(f(I,\varepsilon),x) \geqslant 1 - c(\varepsilon)$ implies $\rho_{\Pi}(I, g(I,\varepsilon,x)) \geqslant 1 - \varepsilon$. ■

This reducibility can be analogously defined when at least one of the involved problems is a minimization problem. Note also that PTAS-reducibility is a generalization of P-reducibility and that *it remains* ***PTAS****-preserving*.

Indeed, the unique difference between these two reducibilities is that in a PTAS-reduction functions f and g depend on ε. This difference is important since this reducibility does not always preserve optimality of solutions. This advantage allows one to prove completeness for polynomially bounded problems.

Really, in [52], it is shown that this is the case of MAX WSAT-PB. This problem is derived from MAX WSAT-*B*: instances and feasible solutions are the same; only the objective function is modified in order that the problem becomes polynomially bounded. If φ is an instance of MAX WSAT-PB on n variables and τ is a truth assignment, then:

$$m(\varphi,\tau) = n + \left\lfloor \frac{n\left(m'(\varphi,\tau) - W\right)}{W} \right\rfloor$$

where m' is the value of τ on φ with respect to MAX WSAT-*B*. The value of every solution for MAX WSAT-PB is now bounded above by $2n$. Every feasible solution being of value at least n, this problem is obviously in **APX**.

Theorem 9 (*[52]*). MAX WSAT-*B PTAS-reduces to* MAX WSAT-PB. *So,* MAX WSAT-PB *is* ***APX****-complete under PTAS-reducibility.*

Finally, let us note that the closure of the class **Max-SNP** under PTAS-reducibility is the **APX** class [53]; in other words, using the notation introduced at the end of Section 4.2, $\overline{\textit{Max}-\textit{SNP}}^{\mathrm{PTAS}} = \mathbf{APX}$.

5.2.2. APX-intermediate problems

We now tackle the existence of intermediate problems for **APX**. Crescenzi and Panconesi show the existence of such problems in **APX**, with respect to **PTAS**, under P-reducibility. This is proved there via a diagonalization principle inspired from that of LADNER for proving the existence of intermediate problems in **NP** [14]. Once more, the problem shown to be intermediate is build ad hoc and is not so natural.

Crescenzi, Kann and Trevisan [48] have pursued these studies and have obtained very interesting results. Under the assumption that polynomial hierarchy is not finite (assumption weaker than **P** ≠ **NP**) they have exhibited a very natural and very well-known **APX**-intermediate problem, the BIN PACKING. This problem is in **APX**; it admits a polynomial time *asymptotic* approximation schema [54] but not a polynomial time approximation schema, if **P** ≠ **NP** (see, for example, [9]).

Let us note that since PTAS-reducibility is a generalization of P-reducibility, the existence of intermediate problems under the former holds also for the latter. Note also that, even if the existence of **NP**-intermediate problems under Karp-reducibility is established by [14], no natural problem is known to have this status. So, the fact that a problem as BIN PACKING is shown to be intermediate for **APX** is interesting per se.

5.2.3. Limits of this approach

The main interest of the results presented until now is that they build a structure for approximability classes. However, they do not make new contributions about inapproximability bounds for the problems proved complete for one or for another class.

As we have already mentioned, the real breakthrough for inapproximability has been the **PCP** theorem. Hence, an a posteriori question is if the advances made thanks to this theorem would be possible via completeness issues. For example, would it be possible to prove the non-existence of a polynomial time approximation schema for MAX 3SAT (and for many other problems for which we know now that they do not belong to **PTAS**) only by proving its **APX**-completeness and without using **PCP**-theorem? Let us note that very recently a combinatorial proof of this theorem has been presented in [38] (see Section 3.3.4); hence inapproximability of, say, MAX 3SAT can be now established by purely combinatorial arguments.

Crescenzi and Trevisan have studied the relationship between the **PCP** theorem and completeness in [55]. They tackle the question whether the **PCP** theorem can be deduced from a proof of the **APX**-completeness of MAX 3SAT (under PTAS-reducibility). They so exhibit a strong link among completeness and **PCP** theorem.

Let us briefly describe the result of [55] by revisiting for a while this theorem. It stipulates that $\mathbf{NP} = \mathbf{PCP}[\mathbf{O}(\log(\mathbf{n})), \mathbf{O}(\mathbf{1})]$. In other words, this reduces the question "$\mathbf{P} = \mathbf{NP}$?" to "$\mathbf{PCP}[\mathbf{O}(\log(\mathbf{n})), \mathbf{O}(\mathbf{1})] = \mathbf{P}$?". They prove that completeness of MAX 3SAT would entail a slightly weaker version of **PCP** theorem expressed by the following theorem.

Theorem 10 (*[55]*). *If* MAX 3SAT *is* **APX***-complete under PTAS-reducibility, then:*

$$\mathbf{PCP}[\mathbf{O}(\log(\mathbf{n})), \mathbf{O}(\mathbf{1})] = \mathbf{P} \Longrightarrow \mathbf{NP} = \mathbf{co-NP}.$$

In other words, **APX**-completeness of MAX 3SAT implies the existence of non-polynomial problems in $\mathbf{PCP}[\mathbf{O}(\log(\mathbf{n})), \mathbf{O}(\mathbf{1})]$ under the assumption $\mathbf{NP} \neq \mathbf{co-NP}$ (instead of $\mathbf{P} \neq \mathbf{NP}$ for **PCP** theorem).

As explained in [55], this result means that any **APX**-completeness proof of MAX 3SAT includes per se a proof (of a slightly weaker version) of the **PCP** theorem. This is also an explanation of why such a completeness result has not been achieved independently.

Finally, let us note that the reciprocal question: "can we deduce completeness from **PCP** theorem?" will be tackled later in Section 7.

5.3. *PTAS-completeness*

PTAS-completeness has been initially tackled by Crescenzi and Panconesi in [51]. They define the following reducibility, called F-reducibility.

Definition 9 (*[51]*). Let Π and Π' be two maximization **NPO** problems. We say that Π reduces to Π' under F-reducibility, if there exist three functions f, g and c such that:

- for every $I \in \mathcal{I}_{\Pi}$, $f(I) \in \mathcal{I}_{\Pi'}$; f is computable in polynomial time;
- for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $g(I, x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; g is computable in polynomial time;
- the complexity of computing c is bounded by $p(1/\varepsilon, |x|)$, where p is a polynomial;
- the value of c is $1/q(1/\varepsilon, |x|)$, where q is a polynomial;
- for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $|1 - \rho_{\Pi'}(f(I), x)| \leqslant c(1/\varepsilon, |x|)$ implies $|1 - \rho_{\Pi}(I, g(I, x, r))| \leqslant \varepsilon$. ■

It is easy to see that *F-reducibility preserves membership in FPTAS*.

Based upon this reducibility, it is shown in [51] that MAX LINEAR WSAT-B is **PTAS**-complete. This problem is a further restricted version of MAX WSAT-B, where the sum of variable-weights is between W and $(1 + (1/n - 1))W$, where n is the number of variables.

MAX LINEAR WSAT-B is obviously in **PTAS**. Given a constant $\varepsilon > 0$, in order to have an $(1 - \varepsilon)$-approximation algorithm, it suffices to return some solution, if $n \geqslant 1/\varepsilon$, or to optimally solve the problem (by exhaustive search) in the case where $n < 1/\varepsilon$.

Theorem 11 (*[51]*). MAX LINEAR WSAT-B *is* **PTAS***-complete under F-reducibility.*

The proof of Theorem 11 is, once more, based upon a modification of Cook's theorem, simulating a particular Turing machine, as the one of Theorem 8. Let us note that MAX LINEAR WSAT-B is the only problem shown to be **PTAS**-complete under F-reducibility. As for **APX**-completeness, the existence of **PTAS**-intermediate problems, i.e., problems that are neither **PTAS**-complete, nor in **FPTAS**, unless $\mathbf{P} \neq \mathbf{NP}$ is shown in [51].

MAX LINEAR WSAT-B is a very artificial problem. On the other hand, F-reducibility itself is very restrictive for allowing the existence of natural **PTAS**-complete problems. More generally, one can wonder if the approach of devising approximation preserving reductions based upon the model of Karp-reducibility (as it is the case of the reducibilities seen until now in this paper) is not intrinsically limited, mainly when dealing with classes "close" to **P** (as **FPTAS**). In this case, we should need a model of basic reducibility more powerful than that of Karp.

The work in [56] proposes a model of approximation preserving reducibility based upon Turing-reducibility rather than upon Karp-reducibility. There, it is assumed that an oracle exists approximately solving the range problem and then, the initial problem is reduced to the range one by deriving an approximation algorithm using this oracle. In other words, [56] proposes a kind of adaptation of Turing-reducibility to the approximation framework. This is reducibility FT.

Definition 10 (*[56]*). Let Π and Π' be two **NPO** problems. Let $\bigcirc_{\alpha}^{\Pi'}$ be an oracle providing, for every instance I' of Π' (and every $\alpha > 0$), a feasible solution x of I', that is an $(1 - \alpha)$-approximation, if $\mathrm{goal}(\Pi') = \max$, an $(1 + \alpha)$-approximation, otherwise. Then Π reduces to Π' under FT-reducibility (denoted by $\Pi \leqslant_{\mathrm{FT}} \Pi'$), if for every $\varepsilon > 0$, there exists an algorithm $\mathcal{A}_{\varepsilon}(I, \bigcirc_{\alpha}^{\Pi'})$ such that:

- for every instance I of Π, $\mathcal{A}_{\varepsilon}$ computes a feasible solution x, that is an $(1-\varepsilon)$-approximation, if $\mathrm{goal}(\Pi) = \max$, an $(1+\varepsilon)$-approximation, otherwise;
- if we assume that, for every instance I' of Π', oracle $\bigcirc_{\alpha}^{\Pi'}(I')$ computes, in polynomial time with $|I'|$ and $1/\alpha$, a solution of I', that is an $(1 - \alpha)$-approximation (resp., an $(1 + \varepsilon)$-approximation), then the execution time of A_{ε} is polynomial with the size of the instance and with $1/\varepsilon$. ■

It is easy to see that FT-reducibility *preserves membership in FPTAS*.

Theorem 12 ([56]). *Let Π be an* **NPO** *problem that has an* **NP**-*complete decision counterpart. If Π is polynomially bounded, then, for any problem $\Pi' \in$* **NPO**, *$\Pi' \leqslant_{FT} \Pi$. Consequently, if an* **NPO-PB** *problem having* **NP**-*complete decision version belongs to* **PTAS**, *then it is* **PTAS**-*complete under FT-reducibility.*

Two classical **NPO** problems, namely MAX PLANAR INDEPENDENT SET and MIN PLANAR VERTEX COVER fit conditions of Theorem 12 (see [57] for their membership in **PTAS**). So, the following theorem is an immediate corollary of Theorem 12.

Theorem 13 ([56]). MAX PLANAR INDEPENDENT SET *et* MIN PLANAR VERTEX COVER *are* **PTAS**-*complete under FT-reducibility.*

FT-reducibility is less restrictive than F-reducibility (in fact the latter is a particular case of the former). This can be shown through Theorem 12 where it is shown, under relatively weak hypotheses, that one can reduce an **NPO** problem to one belonging to **PTAS**. Such a result can be also expressed, in terms of a closure property: $\overline{\mathbf{PTAS}}^{\mathrm{FT}} = \mathbf{NPO}$.

Dealing with FT-reducibility another question arises, namely the existence of intermediate problems (indeed, the less restrictive a reducibility, the more unlikely the existence of such problems). The principal difficulty for studying this question, is the nature of FT-reducibility itself. Recall that, as already mentioned, it is closer to Turing-reducibility than to Karp-reducibility. It comes out that it is difficult to follow Ladner's approach [14] and to use a carefully designed diagonalization technique. On the other hand, to our knowledge, the existence of **NP**-intermediate problems (assuming $\mathbf{P} \neq \mathbf{NP}$) under Turing-reducibility still remains open.

Face to these difficulties, one can study the question of **PTAS**-intermediate problems (for FT-reducibility) under another (weaker) hypothesis than $\mathbf{P} \neq \mathbf{NP}$. Let us observe that the optimization-version of Turing-reducibility preserves membership in **PO**. Then, as it is proved in [56], the existence of **NPO**-intermediate problems (with respect to **PO**) under the optimization version of Turing reducibility, is a sufficient condition for the existence of **PTAS**-intermediate problems under FT-reducibility. This is stated in the following theorem.

Theorem 14 ([56]). *If there exists an* **NPO**-*intermediate problem (with respect to* **PO**) *under Turing-reducibility, there exists a* **PTAS**-*intermediate problem under FT-reducibility.*

6. Max-SNP, Max-NP and L-reducibility

We have already seen in Section 3.4 the links between logic, complexity and approximation via the definition of syntactic classes **Max-SNP** and **Max-NP** in [40]. Both of them are subclasses of **APX** and include many very natural and well-known problems. Indeed, with their work, Papadimitriou and Yannakakis perform the first systematic approach for apprehending completeness in approximation classes.

The first step in [40] is to define a reducibility preserving membership in **PTAS**; this is the well-known L-reducibility already presented in Section 4.1. The great advantage of **Max-SNP** and **Max-NP** is that their problems are defined in a much more structured way than problems in **NPO**. Thanks to this, authors in [40] devise a generic L-reduction leading to the following result.

Theorem 15 ([40]). MAX 3SAT *is* **Max-SNP**-*complete under L-reducibility.*

This is a fundamental result. Indeed, it reduces the existence of polynomial time approximation schemata for the whole class **Max-SNP** to the existence of a polynomial time approximation schema for MAX 3SAT. This class containing a large number of problems, the result of Theorem 15, even if it does not constitute a proof of the absence of polynomial time approximation schema for MAX 3SAT gives, nevertheless, strong evidence about this fact.

Corollary 3 ([40]). MAX 3SAT *is in* **PTAS** *if and only if* $\mathbf{Max-SNP} \subseteq \mathbf{PTAS}$.

Moreover, the completeness claimed by Theorem 15 distances itself from the ones seen in Section 5 by the way it is obtained. It is not based upon another adaptation of the proof of Cook's theorem but rather upon a transformation of logical formulæthat has become possible thanks to the careful definition of **Max-SNP**.

In [40] a lot of other problems are shown **Max-SNP**-complete as MAX INDEPENDENT SET-B, or MAX CUT. But if such results have been possible for **Max-SNP** things were different for **Max-NP** for which no such results appear in [40]. Indeed, the existence of complete problems for **Max-NP** is mentioned as an open problem in [40]. Such a result, for instance, would deepen the knowledge on the relationship between **Max-SNP** and **Max-NP**. Obviously, $\mathbf{Max-SNP} \subset \mathbf{Max-NP} \subset \mathbf{APX}$. This last inclusion made researchers think that structure of **Max-NP** problems is potentially richer than that of **Max-SNP** problems and that it is not harder to approximately solve a problem of **Max-NP** than a problem of **Max-SNP**. So the question is "are all the problems of **Max-NP** reducible to a problem of **Max-SNP**?". Let us note that answering positively to this question would allow us to directly answer to the question of existence of **Max-NP**-complete problems.

The answers come from Crescenzi and Trevisan [55]. Coming up against the somewhat restrictive character of L-reducibility, they study the completeness of **Max-NP** under PTAS-reducibility (Definition 8, Section 5.2), reducibility smoother than the L-reducibility. In this way they prove that MAX 3SAT *is* **Max-NP**-*complete under PTAS-reducibility*. Let us remark that, as we will see in detail in Section 7, this result was already known. Indeed, thanks to the **PCP** theorem, it can be shown that MAX 3SAT is **APX**-complete under PTAS-reducibility (see also Section 7.2). **Max-NP** being included in **APX**, **Max-NP**-completeness of MAX 3SAT is a corollary of its **APX**-completeness.

In fact, the interest of the result of [55] lies in its proof that *does not use the PCP theorem*. This proof uses a direct generic reduction structurally transforming problems of **Max-NP** into problems of **Max-SNP** and this reduction has the advantage of being constructive. The instance of MAX 3SAT is explicitly specified.

Finally, let us note that MAX SAT, as well as any **Max-SNP** complete problem under L-reducibility, is **Max-NP**-complete under PTAS-reducibility.

7. Completeness using PCP theorem

Summarizing briefly the results presented in Sections 5 and 6, we could remark on the following:

- by using proofs inspired by the proof of Cook's theorem and approximation preserving reducibilities based upon Karp-reducibility, we can obtain complete problems for the main "combinatorial" approximation classes (such as **APX** and **PTAS**) but these problems are not natural;
- by defining syntactic classes, we obtain complete problems for a natural subclass of **APX**; however, the tools developed in this framework do not allow us to obtain completeness for the whole class **APX**.

The **PCP** theorem has provided (negative) answers to approximability of an important number of natural and well-known problems. Among them, MAX 3SAT, a central problem for **Max-SNP**. So, could we use the powerful system of probabilistic checkable proofs in order to achieve new completeness results? Can we, thanks to this system (or, rather, to these systems), establish new generic reductions?

Khanna, Motwani, Sudan and Vazirani [53] give pertinent answers to these questions, exhibiting new links and providing new insights to links between completeness and inapproximability. They show how an inapproximability result obtained by **PCP** theorem can be transformed into a completeness result.

Furthermore, the scope of the method developed by [53] is quite general and also applies to approximation classes beyond **APX**. Indeed, a completeness result for an approximation class can be seen as an instantiation of this method to the class and the problem under consideration. They so obtain completeness results for several classes and bring a definite (positive) answer to the central question of completeness of MAX 3SAT for **APX**.

7.1. *From PCP theorem to completeness*

Faced with the apparent impossibility to get completeness results for combinatorial classes using L-reducibility, in [53] a slightly weaker reducibility, called E-reducibility is introduced.

Definition 11 ([53]). Let Π and Π' be two minimization problems of **NPO**. We say that Π reduces to Π' under E-reducibility, if there exist two polynomial time computable functions f et g, a constant β and a polynomial p such that:

1. for every $I \in \mathcal{I}_{\Pi}$, $f(I) \in \mathcal{I}_{\Pi'}$; furthermore, $\mathrm{opt}_{\Pi'}(f(I)) \leqslant p(|I|)\mathrm{opt}_{\Pi}(I)$;
2. for every $I \in \mathcal{I}_{\Pi}$ and every $x \in \mathrm{Sol}(f(I))$, $g(I,x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; furthermore, $\rho_{\Pi}(I, g(I,x)) - 1 \leqslant \beta(\rho_{\Pi}(f(I),x) - 1)$. ■

E-reducibility can be analogously defined if at least one of Π and Π' is a maximization problem. Let us note also that condition 2 in Definition 11, dealing with transformation of approximation ratios, is quite similar to the corresponding one of L-reducibility (Definition 3). The main difference between the two reducibilities is that E-reducibility relaxes the linear dependence between optima, imposed in a L-reduction, to a polynomial one. Also, E-reducibility *preserves membership in* **PTAS**.

Let us now tackle the relationship between inapproximability and completeness. Let us first note that inapproximability results achieved using **PCP** theorem can be rewritten in a particular form called *canonical hardness*.

Let us recall some basic definitions of [53]. Consider a family $\mathcal{F}$ of functions from $\mathbb{N}$ to $\mathbb{N}$. Then, $\mathcal{F}$ is said to be *downward close* if, for every function $g \in \mathcal{F}$ and every constant c, $h(n) = O(g(n^c))$ implies $h \in \mathcal{F}$. A function g is said to be *hard* for $\mathcal{F}$ if, for every $h \in \mathcal{F}$, there exists a constant c such that $h(n) = O(g(n^c))$. If, furthermore $g \in \mathcal{F}$, then it is said to be *complete*. Denote by $\mathcal{F}$-**APX** (resp., $\mathcal{F}$-**APX-PB**), the class of problems (resp., polynomially bounded problems) approximable within standard-ratio $g(n)$ for some $g \in \mathcal{F}$.

Remark that the class of functions bounded above by a constant, or by a logarithmic function, or even, by a polynomial are downward close classes. These classes correspond to approximation classes **APX**, **Log-APX** and **Poly-APX**, respectively. Note furthermore that constant, or logarithmic, or polynomial functions are complete for the three corresponding classes.

A maximization **NPO** problem Π is said to be *canonically* hard for the class $\mathcal{F}$-**APX**, for a family $\mathcal{F}$ of downward close functions, if there exist a function f, computable in polynomial time, two constants n_0 and c and a function G, hard for $\mathcal{F}$, such that:

1. for every instance φ of 3SAT on $n \geqslant n_0$ variables, and for every $N \geqslant n^c$, $f(\varphi, N)$ is an instance of Π;
2. if φ is satisfiable, then $\mathrm{opt}(f(\varphi,N)) = N$;
3. if φ is not satisfiable, then $\mathrm{opt}(f(\varphi,N)) = N/G(N)$;
4. given a solution x of $f(\varphi,N)$ of value (strictly) greater $N/G(N)$, we can compute in polynomial time a truth assignment satisfying φ.

As mentioned in [53], one can analogously define canonically hard minimization problems by replacing in item 2 above $N/G(N)$ by $NG(N)$ and by accordingly modifying items 3 et 4.

Since 3SAT is **NP**-complete, an equivalent definition of canonical hardness can be the following: *a problem is canonically hard if every problem of* **NP** *can be reduced to it following rules* 1 to 4. Following this remark, the notion of canonical hardness appears to be very close to GAP-reducibility. In fact, a proof of canonical hardness is a kind of GAP-reducibility working for a family of functions, rather than for a particular ratio's value, considering that we can find a polynomial certificate proving that instance is positive (when this is the case). This slight difference is crucial for the purposes of [53].

The interest of the concept of canonical hardness is that it fits well results implied by the **PCP** theorem. These results provide a family of reducibilities slightly more powerful (but of the same spirit and type) than the GAP-reducibility. However, to get the completeness results that are of interest for us, reductions among optimization problems are needed. Henceforth, we must use the power of reductions provided by the inapproximability corollaries of the **PCP** theorem in order to devise reductions between optimization problems.

Khanna et al. have surrounded this difficulty by devising a reduction from an **NPO** problem Π to a canonically hard (for some approximation class) problem Π', using for this purpose reductions from a set of decision problems associated with Π to Π'.

Before describing the idea of the method of [53] we introduce a last notion, the one of an *additive problem*. A problem $\Pi \in$ **NPO** is said to be *additive* if there exist two functions $\oplus$ et f, computable in polynomial time, such that:

- for any pair (I_1, I_2) of instances of Π, $I_1 \oplus I_2$ is an instance of Π such that $\mathrm{opt}(I_1 \oplus I_2) = \mathrm{opt}(I_1) + \mathrm{opt}(I_2)$;
- for any solution x of $I_1 \oplus I_2$, $f(x)$ is a pair (x_1, x_2) of solutions for I_1 and I_2, respectively, such that $m(I_1 \oplus I_2, x) = m(I_1, x_1) + m(I_2, x_2)$.

The following theorem introduces a very narrow relation between canonical hardness and completeness in approximation classes. It constitutes the stepping stone of the link among inapproximability via the **PCP** theorem and completeness.

Theorem 16 ([53]). *If $\mathcal{F}$ is a downward close family, then any problem Π additive and canonically hard for $\mathcal{F}$-**APX**, is $\mathcal{F}$-**APX-PB**-hard under E-reducibility.*

7.2. APX-completeness

The first application of the **PCP** theorem to approximation provides, as we have already mentioned, an inapproximability result for MAX 3SAT. Starting from an instance I of a problem $\Pi \in$ **NP**, one builds, via the **PCP** theorem, an instance φ of MAX 3SAT such that [30,53]:

- if I is positive, then φ is satisfiable;
- if I is negative, then at most a fraction $(1 - \varepsilon)$ of clauses of φ are satisfiable;
- if a truth assignment satisfies strictly more than a fraction $(1 - \varepsilon)$ of clauses of φ, we can recover in polynomial time a certificate proving that I is positive.

Theorem 17 ([30]). MAX 3SAT *is canonically hard for **APX-PB**.*

Combining Theorems 16 and 17, we get the following fundamental result from [53].

Theorem 18 ([53]). MAX 3SAT *is **APX-PB**-complete under E-reducibility.*

This result is fundamental because, on one hand, it is the first completeness result established via the use of the **PCP** theorem and, on the other hand, it establishes the completeness of a paradigmatic problem for a natural combinatorial approximation class. However, this completeness is not established for the whole class **APX**. For doing this it suffices to consider not E-reducibility, quite restrictive for such a result, but PTAS-reducibility.

Theorem 19 ([53]). MAX 3SAT *is **APX**-complete under PTAS-reducibility.*

Starting from Theorem 19 completeness of many other problems can be established. In [40] numerous problems have been shown **Max-SNP**-complete under L-reducibility (as, for instance, MAX INDEPENDENT SET-B, or MAX CUT). All these problems become **APX**-complete under PTAS-reducibility. Nowadays, a lot of problems are known to be **APX**-complete. For more about them, the interested reader can refer to [4].

7.3. Completeness beyond APX

As underlined above, one of the major interests of [53] is that the authors exhibit a generic link between inapproximability and completeness, applying to many approximation classes. In what follows we present completeness results concerning two other well-known approximation classes corresponding to two natural families of downward close functions: the one bounded by a polynomial and the one bounded by a logarithmic function.

7.3.1. Completeness in Poly-APX-PB and Log-APX-PB

Using interactive proof systems, it is shown in [29] that there exists some $c > 0$ such that MAX INDEPENDENT SET is not approximable within approximation ratio n^{-c}, unless **P** = **NP**. This result has been strengthened by using optimized **PCP** systems to get $c = 1/2$ [23].

Following terminology and concepts from [53], the above results can be formulated as follows.

Theorem 20 ([29,23]). MAX INDEPENDENT SET *is canonically hard for **Poly-APX-PB**.*

Combining Theorems 16 and 20, the following theorem is directly derived.

Theorem 21 ([53]). MAX INDEPENDENT SET *and* MAX CLIQUE *are **Poly-APX-PB**-complete under E-reducibility.*

Later developments inspired from [53] have provided a strong inapproximability result for MIN SET COVER, known to be approximable within $O(\log n)$ [1,58,59]. Raz and Safra [60] claim that it is hard to approximate MIN SET COVER within better than $c \ln(n)$ for some constant $c > 0$.

Based upon what has been discussed just previously, the following result can be derived.

Theorem 22 ([53,60]). MIN SET COVER *is canonically hard for **Log-APX-PB**. Consequently,* MIN SET COVER *is **Log-APX-PB**-complete under E-reducibility.*

7.3.2. Completeness in Poly-APX

Unfortunately, if we try to generalize the result of Theorem 21 to the whole class **Poly-APX** we are faced with a major difficulty. Revisit for a while the case of **APX**-completeness. We have mentioned in Section 5.2 that it is very unlikely that one could prove **APX**-completeness of a polynomially bounded problem under a reducibility preserving optimality since this would imply $\mathbf{P}^{\mathrm{SAT}} = \mathbf{P}^{\mathrm{SAT}[O(\log(n))]}$. This fact has motivated introduction of reducibilities that did not preserve optimality, as PTAS-reducibility. Starting from this remark, it is also very unlikely to get a result of **Poly-APX**-completeness for MAX INDEPENDENT SET under E-reducibility.

In [56] the possibility is studied to get **Poly-APX**-completeness not via E-reducibility, but rather via the less restrictive PTAS-reducibility. The parallel between canonical hardness and completeness is then expressed as follows.

Theorem 23 ([56]). *If $\Pi' \in$ **NPO** is a maximization problem additive, and canonically hard for **Poly-APX**, then any maximization problem in **Poly-APX** PTAS-reduces to Π'.*

Let us note that if $\mathrm{goal}(\Pi) = \min$, one can PTAS-reduce it to a maximization problem of **Poly-APX**, as indicated in [53].

We so have a completeness result for the whole class **Poly-APX**. Using then canonical hardness for **Poly-APX** of MAX INDEPENDENT SET and MAX CLIQUE, we immediately get the following theorem.

Theorem 24 *([56]).* MAX INDEPENDENT SET *and* MAX CLIQUE *are* ***Poly-APX****-complete under PTAS-reducibility.*

7.3.3. Completeness in **Log-APX**

The observation made for **Poly-APX** in the beginning of Section 7.3.2, remains valid for **Log-APX** too. Hence, if one wishes to study completeness for the whole **Log-APX** she/he must do it via reducibilities "larger" than E-reducibility.

In [61] a slight modification of PTAS-reducibility is introduced. This new reducibility, called MPTAS, is defined as follows.

Definition 12 ([61]). Let Π and Π' be two maximization problems of **NPO** (case of minimization is completely analogous). We say that Π reduces to Π' under MPTAS-reducibility, if and only if there exist two polynomial time computable functions f et g, and a function c such that:

- for every $I \in \mathcal{I}_\Pi$ and every $\varepsilon \in]0,1[$, $f(I,\varepsilon) = (I'_1, I'_2, \ldots, I'_M)$ is a family of instances of Π' (where M is polynomially bounded with $|I|$);
- for every $I \in \mathcal{I}_\Pi$, every $\varepsilon \in]0,1[$ and every family $x = (x_1, x_2, \ldots, x_M)$ of feasible solutions, where x_i is a feasible solution of I'_i, $g(I,x,\varepsilon) \in \mathrm{Sol}(I)$;
- $c :]0,1[\rightarrow]0,1[$;
- there exists some j such that, for all $I \in \mathcal{I}_\Pi$ and $\varepsilon \in]0,1[$, $\rho_{\Pi'}(I'_j, x_j) \geqslant 1 - c(\varepsilon)$ implies $\rho_\Pi(I, g(I,x,\varepsilon)) \geqslant 1 - \varepsilon$. ■

It is easy to see that MPTAS-reducibility preserves membership in **PTAS**. Also, the fact that function f in Definition 12 is multivalued relaxes restriction to additive problems and applies even to non-additive ones.

Theorem 25 *([61]). Let $\mathcal{F}$ be a family of downward close functions and $\Omega \in$* **NPO** *a maximization problem canonically hard for* $\mathbf{F-APX}$*. Then, any maximization problem in* **NPO** $\cap$ $\mathbf{F-APX}$ *reduces to* Ω *under MPTAS-reducibility.*

On the other hand, it can be shown as in [53], that any minimization problem of $\mathbf{F-APX}$ E-reduces (hence, MPTAS-reduces too) to a maximization problem of $\mathbf{F-APX}$, getting so the following generalization of Theorem 25.

Theorem 26 *([61]). Let $\mathcal{F}$ be a downward close family of functions and $\Omega \in$* **NPO** *a canonically hard problem for* $\mathbf{F-APX}$*. Then, any problem in* **NPO** $\cap$ $\mathbf{F-APX}$ *reduces to* Ω *under MPTAS-reducibility.*

Consider now class **Log-APX**. MIN SET COVER is approximable within ratio $O(\log n)$ [59]; hence it belongs to **Log-APX**. Furthermore, as we have already mentioned in Section 7.3.1, it is inapproximable within a ratio smaller than $c \log n$, for some constant c, unless $\mathbf{P} = \mathbf{NP}$ (see [62,60] where this result is mentioned, as well as [61] for an informal proof). Applying Theorem 26, the following result can be derived.

Theorem 27 *([61]).* MIN SET COVER *is* ***Log-APX****-complete under MPTAS-reducibility.*

8. Completeness in differential approximation

The study of approximation following the differential paradigm has been developed, as we have mentioned, mainly at the beginning of the 90's. After a first paper operationally and mathematically justifying the use of the differential ratio [17], a systematic study of differential approximation of **NPO** problems has started and still continues. Several results (positive or negative) for classical combinatorial problems have appeared (MIN COLORING [21,63,64], MIN TSP, MAX TSP and vehicle routing problems [65–68], BIN PACKING [69,70], MIN SET COVER [71], optimal satisfiability problems [24,72], etc.). Several structural and computational aspects are also investigated in [73–76].

Naturally, in a second time, structure in differential approximation classes has also been tackled. If notions of reducibility well-adapted to this paradigm have appeared quite early, in particular the affine reducibility (see Definition 4 in Section 4.1), completeness results have been obtained recently [77,56].

Obviously, the study of the **NPO** structure for the differential paradigm is modeled on the standard one. So, this study captures completeness for **NPO**, **DAPX**, **DPTAS**, **Poly-DAPX** (recall that no natural problems are still known to be in **Log-DAPX**, or in **Exp-DAPX**), as well as for the class **0-DAPX**, a class proper to differential paradigm (see [24] and Section 3.2). These classes are tackled in what follows.

8.1. Completeness in NPO

Following the work of Orponen and Manilla [43], the following reducibility, called D-reducibility (the differential counterpart of strict-reducibility) is defined in [77].

Definition 13 ([77]). Let Π and Π' be two **NPO** problems. We say that Π reduces to Π' under D-reducibility, if there exist two functions f and g, computable in polynomial time, such that:

- for every $I \in \mathcal{I}_\Pi$, $f(I) \in \mathcal{I}_{\Pi'}$;
- for every $I \in \mathcal{I}_\Pi$ and every $x \in \mathrm{Sol}(f(I))$, $g(I,x) \in \mathrm{Sol}(\mathcal{I}_\Pi)$;
- for every $I \in \mathcal{I}_\Pi$ and every $x \in \mathrm{Sol}(f(I))$, $\delta_\Pi(I, g(I,x)) \geqslant \delta_{\Pi'}(f(I), x)$. ■

Obviously, *D-reducibility preserves membership in* ***DAPX*** *and* ***DPTAS***.

By an approach similar to the one in the standard-approximation paradigm, i.e., a combination of proofs of partial completeness (in **Max-NPO** and **Min-NPO**) and differential equivalence (under D-reducibility) between MIN WSAT and MAX WSAT, the following completeness result can be obtained.

Theorem 28 *([77]).* MIN WSAT, MAX WSAT, MIN LINEAR PROGRAMMING *0-1 and* MAX LINEAR PROGRAMMING *0-1 are* ***NPO****-complete under D-reducibility.*

8.2. NPO-completeness and 0-DAPX

As already mentioned in Section 3.2, **0-DAPX** is the class of problems for which any polynomial time algorithm returns a worst solution on at least one instance of them. In other words, for the problems in **0-DAPX**, their differential approximation ratio is equal to 0. In some sense, this class contains the hardest **NPO** problems to approximately solve under the differential paradigm. On the other hand, the notion of completeness itself reflects the hardest problems for a class. So, the following question arises: *"what is the relation between* **0-DAPX** *and* **NPO***-complete problems under D-reducibility?"*. The following theorem brings the first answer to his question.

Theorem 29 ([77]). *Under D-reducibility,* **NPO**−complete ⊆ **0-DAPX**. *In other words, every* **NPO***-complete problem belongs to* **0-DAPX**.

Theorem 29 seems to confirm the idea that the preorder between problems induced by some natural reducibility models a notion of hardness that is real and computational and not only structural and theoretical.

We also note that, if instead of D-reducibility, a somewhat stronger reducibility is used, for instance, if f and g in Definition 13 are multivalued, then, under such reducibility, the class of **NPO**-complete problems coincides with **0-DAPX** [77].

8.3. *Completeness in DAPX*

As we have seen in Sections 5–7, **APX** is the most important standard-approximation class and the one that has motivated and mobilized the most numerous of the studies about its structure and the existence of complete or intermediate problems.

It appears quite natural that the same holds for its differential counterpart, the class **DAPX**. Obviously, the first step consists of carefully defining some notion of reducibility, "smoother" than AF-, or D-reducibility, that preserves membership in **DPTAS**. In [77] the following reducibility, quite similar to PTAS-reducibility is defined.

Definition 14 ([77]). Let Π and Π' be two **NPO** problems. We say that Π reduces to Π' under DPTAS-reducibility, if there exist three functions f, g et c such that:

1. for every $I \in \mathcal{I}_{\Pi}$ and every $\varepsilon \in]0,1[$, $f(I,\varepsilon) \in \mathcal{I}_{\Pi'}$; f is computable in time polynomial with $|I|$;
2. for every $I \in \mathcal{I}_{\Pi}$, every $\varepsilon \in]0,1[$ and every $x \in \mathrm{Sol}(f(I))$, $g(I,\varepsilon,x) \in \mathrm{Sol}(\mathcal{I}_{\Pi})$; g is computable in time polynomial with $|I|$ and $|x|$;
3. $c :]0,1[\rightarrow]0,1[$;
4. for every $I \in \mathcal{I}_{\Pi}$, every $x \in \mathrm{Sol}(f(I))$ and every $\varepsilon \in]0,1[$, $\delta_{\Pi'}(f(I,\varepsilon),x) \geqslant 1-c(\varepsilon)$ implies $\delta_{\Pi}(I,g(I,\varepsilon,x)) \geqslant 1-\varepsilon$;
5. function f can be multivalued; in this case $f = (f_1,\ldots,f_i)$, where i is polynomial with $|I|$; in this case item 4 becomes: there exists $j \leqslant i$ such that $(\delta_{\Pi'}(f_i(I,\varepsilon),x) \geqslant 1-c(\varepsilon)$ implies $\delta_{\Pi}(I,g(I,\varepsilon,x)) \geqslant 1-\varepsilon)$. ■

Once more, DPTAS-*reducibility preserves membership in* **DPTAS**.

In order to apprehend the existence of complete problems for **DAPX**, the first idea could be to adapt (if possible) the proof of [51] in the differential context, with the risk (in the case that it worked) to produce non-natural complete problems. A potentially more fruitful approach is to use the **PCP** theorem. The problem is that this theorem is very closely connected to standard approximation and it seems very difficult to adapt it in the differential framework in order to get completeness.

The reasoning schema developed in [77] tries to exploit completeness results in **APX** (using so indirectly the **PCP** theorem). It can be summarized as follows. For proving that a (first) problem Π_1 is **DAPX**-complete:

- one first searches a problem Π_2 which is **APX**-complete but that also reduces to Π_1 under an ad hoc reduction that transforms a polynomial time differential-approximation schema into a polynomial time standard-approximation schema;
- next, starting from a problem $\Pi_3 \in$ **DAPX**, one reduces it to a problem $\Pi_4 \in$ **APX** under another ad hoc reduction that this time transforms a polynomial time standard-approximation schema into a polynomial time differential-approximation schema;
- we finally obtain, by transitivity, a reduction of any problem $\Pi_3 \in$ **DAPX** to Π_1.

Putting all the above together, an eventual polynomial time differential-approximation schema for Π_1 provides a polynomial time standard-approximation schema for Π_2, hence a polynomial time standard-approximation schema for Π_4, hence a polynomial time differential-approximation schema for Π_3. In other words, $\Pi_3 \leqslant \Pi_4 \leqslant \Pi_2 \leqslant \Pi_1$ (the reductions appearing in this expression being different the ones from the others). Using this reasoning schema, the following theorem can be obtained.

Theorem 30 ([77]). MAX INDEPENDENT SET-*B*, MIN VERTEX COVER-*B*, MAX SET PACKING-*B and* MIN SET COVER-*B are* **DAPX**-*complete under* DPTAS*-reducibility.*

For instance, MAX INDEPENDENT SET-*B* is **APX**-complete and, furthermore, for this problem, standard- and differential-approximation ratios coincide. So, the main part of the work for the proof of Theorem 30, consists of showing that every problem of **DAPX** can be reduced to a problem of **APX** under a reducibility transforming a polynomial time standard-approximation schema into a polynomial time differential-approximation schema. This leads to a proof of **DAPX**-completeness of MAX INDEPENDENT SET-*B*. Completeness of the other problems stated in Theorem 30 is obtained in [77] by DPTAS-reductions from MAX INDEPENDENT SET-*B*.

So, many natural problems are now known to be **DAPX**-complete showing that completeness in the differential paradigm is as pertinent and interesting as for the standard paradigm. However, at the moment where [77] was achieved, one could notice the following two dampers for Theorem 30:

1. no intermediate problems are known for **DAPX** under DPTAS-reducibility;
2. problems stated in Theorem 30 are all **APX**-complete; it would be interesting to find problems that are **DAPX**-complete but not **APX**-complete.

Point 1 remains still open and deserves further research. For point 2, it is shown in [56] that MIN COLORING is **DAPX**-complete, under DPTAS-reducibility, while it does not belong to **APX**.

8.4. *Completeness in DPTAS*

The existence of **DPTAS**-complete problems has been initially tackled in [77], using Karp-type reducibilities, but with very partial results mainly concerning subclasses of **DPTAS**. In [56], the approach described in Section 5.3 has been also extended to the differential paradigm.

Indeed FT-reducibility (Definition 10, Section 5.3) is quite large and can be re-expressed to fit the differential approximation also. Let us denote by DFT the differential counterpart of FT-reducibility. It can be immediately seen that *DFT-reducibility preserves membership in* **DFPTAS**. So, the following theorem holds.

Theorem 31 (*[56]*). *Let $\Pi \in$ **NPO** be a problem having **NP**-complete decision version. If Π is differentially polynomially bounded, then any problem $\Pi' \in$ **NPO** reduces to Π under DFT-reducibility. Consequently, if a differentially polynomially bounded **NPO** problem having **NP**-complete decision version belongs to **DPTAS**, then it is **DPTAS**-complete under DFT-reducibility.*

It suffices now to remark that for MAX PLANAR INDEPENDENT SET standard and differential ratios coincide and to recall that MAX PLANAR INDEPENDENT SET $\in$ **PTAS** [57]. On the other hand, MIN PLANAR VERTEX COVER belongs also to **DPTAS**, by an AF-reduction to MAX PLANAR INDEPENDENT SET. Finally, BIN PACKING belongs to **DPTAS** [69]. All these problems being differentially polynomially bounded **NPO** problems having **NP**-complete decision versions, the following theorem is immediately derived.

Theorem 32 (*[56]*). MAX PLANAR INDEPENDENT SET, MIN PLANAR VERTEX COVER *and* BIN PACKING *are **DPTAS**-complete under DFT-reducibility.*

We finally note that the same result dealing with conditions of existence of intermediate problems (the differential counterpart of Theorem 14) holds also for DFT-reducibility.

8.5. *Completeness in Poly-DAPX*

We conclude this state of the art on the intersection of complexity theory and polynomial approximation theory by tackling completeness in **Poly-DAPX**. This question is studied in [56], by using DPTAS-reducibility. In any case, the problem of using reducibilities preserving optimality or not, discussed in Section 7.3.2, is posed under the same terms in the differential paradigm also.

Let us also note that dealing with the differential paradigm, one can easily restrict her/himself to maximization problems. Indeed, when dealing with a minimization problem Π, one can define a maximization problem Π' having the same set of instances and of feasible solutions with Π and with objective function $m_{\Pi'}(I, x) = M - m_{\Pi}(I, x)$ (where M is an upper bound of the value of the solutions of instance I). Problems Π and Π' are affine-equivalent (so, in particular, Π DPTAS-reduces to Π'). Furthermore, if $\Pi \in$ **Poly − DAPX**, then $\Pi' \in$ **Poly − DAPX**.

Theorem 33 (*[56]*). *If Π is canonically hard for **Poly-DAPX**, then any problem in **Poly-DAPX** DPTAS-reduces to Π.*

One can notice that in the statement of Theorem 33, the additivity of Π is omitted, contrary to the case of Theorem 23 (Section 7.3.2) dealing with standard approximation and PTAS-reducibility. This is due to the fact that a DPTAS-reduction can be multivalued. We could also relax additivity in Theorem 23 if we allowed a PTAS-reduction to be multivalued.

Given that for MAX INDEPENDENT SET and MAX CLIQUE standard- and differential-approximation ratios coincide, that both of them belong to **Poly-APX** (Section 7.3.2), and that MIN VERTEX COVER is affine-equivalent to MAX INDEPENDENT SET, the following result immediately holds.

Theorem 34. MAX INDEPENDENT SET, MAX CLIQUE *and* MIN VERTEX COVER *are **Poly-DAPX**-complete under DPTAS-reducibility.*

9. Discussion

The research program that aims at transposing notions of reducibility and completeness, concepts originally devised for decision problems, to optimization problems, is a very extensive program, active for over thirty years. Numerous notions of reducibilities have been defined and a lot of results have been achieved using them. Despite the scientific interest of all of them, an exhaustive presentation was impossible. This survey has just presented those that have played a key-role for creating a structure (providing completeness results) for **NPO**. For instance, we have not mentioned the interesting notion of *continuous reducibility* presented in [78].

We have seen that generic reductions that structure approximability classes have initially been based upon tricky modifications of the proof of Cook's theorem. They have derived interesting though somewhat partial results, not reaching answers to fundamental questions of approximation theory. In particular they have not been able to prove completeness for **APX** of paradigmatic problems as MAX 3SAT, that plays in approximation theory the role that SAT, or 3SAT, play in classical complexity theory.

The great tool that the **PCP** theorem has brought to approximation and the fantastic advances performed thanks to it and to its subsequent improvements and corollaries, has allowed one to strengthen the links between completeness and inapproximability, providing a generic method that allows to achieve completeness results starting from inapproximability ones. The role of the work by Khanna et al. [53] has been conclusive for this. Trevisan [7] has commented this spectacular advance writing that Khanna et al. *"have provided a definite answer to the question of completeness in approximation classes"*. Indeed, the research programme on approximability preservation and completeness has been so successful that today natural problems are known to be complete for all the standard-approximation classes and for most of the differential ones.

Results presented in this survey concern approximation classes where approximation levels are constant or functions of the instance-size. However, one can consider completeness notions for other approximation classes. For instance, Ausiello and Protasi define in [79] a reducibility preserving not only approximation but also local optimality (the same concept is also marginally considered by [53]). Under this reducibility, they obtain completeness results for a class of problems admitting "good (guaranteed)" local optima. It would be interesting that such structural studies the beginnings of which are presented in [77,74,76] are undertaken also in a differential paradigm.

Another issue deserving further research, is about the refinement and a better apprehension of E-reducibility for determining if it allows the existence of intermediate problems or not. In fact, the only class for which we know such problems (under E-reducibility) is the class **APX**. In the same spirit, F-reducibility, has the merit of being the first that has introduced **PTAS**-completeness [51]. However, as mentioned previously, no natural problem has been proved **PTAS**-complete under it. Is it possible to show the existence of such problems using this reducibility?

Also, and this is, to our opinion the sense of Trevisan's comment, we remark that generic reductions based upon either adaptations of Cook's theorem, or **PCP** theorem, have not brought any additional information on inapproximability of the problems implied. They present a real structural interest, they show strong links between optimization problems but the inapproximability implied by these completeness results were already known. This fact seems to be intrinsic to reductions based upon **PCP** theorem but it is more "disappointing" for reductions based upon Cook's theorem.

In the same spirit, results around **Max-SNP** appear to be very singular. Indeed, L-reductions have two very interesting peculiarities. First, it is "more constructive" than the other reductions seen in this paper. Second, as we have already mentioned, **Max-SNP**-completeness of MAX 3SAT really follows the lines of **NP**-completeness proofs: even if it does not directly provide a definite answer to the existence of a polynomial time approximation schema for this problem, it gives, however, a strong evidence about its non-existence, linking the existence of such schema to the existence of polynomial time approximation schemata for any other **Max-SNP** problem. When this result has been produced, it has represented spectacular advances about the approximability of MAX 3SAT.

Starting from this assessment, we can wonder whether an approach based upon logical definition of **NPO** problems as those appearing, for instance, in [80,81,46] but not presented here, could not enrich completeness issues and perspectives and bring new results and insights.

Acknowledgements

The very pertinent and useful suggestions of two anonymous referees are gratefully acknowledged.

REFERENCES

[1] D.S. Johnson, Approximation algorithms for combinatorial problems, J. Comput. System Sci. 9 (1974) 256–278.
[2] S.A. Cook, The complexity of theorem-proving procedures, in: Proc. STOC'71, 1971, pp. 151–158.
[3] R.M. Karp, Reducibility among combinatorial problems, in: R.E. Miller, J.W. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, New York, 1972, pp. 85–103.
[4] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation. Combinatorial Optimization Problems and their Approximability Properties, Springer-Verlag, Berlin, 1999.
[5] G. Ausiello, V.Th. Paschos, Reductions, completeness and the hardness of approximability, European J. Oper. Res. 172 (2006) 719–739. Invited review.
[6] P. Crescenzi, A short guide to approximation preserving reductions, in: Proc. Conference on Computational Complexity, 1997, pp. 262–273.
[7] L. Trevisan, Inapproximabilité des problèmes d'optimisation combinatoire, in: V.Th. Paschos (Ed.), Optimisation combinatoire: concepts avancés, Hermès Science, Paris, 2005, pp. 91–138. French translation of [82] by B. Escoffier (chapter 3).
[8] V. Vazirani, Approximation Algorithms, Springer, Berlin, 2001.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-completeness, W. H. Freeman, San Francisco, 1979.
[10] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.
[11] C.H. Papadimitriou, K. Steiglitz, Combinatorial Optimization: Algorithms and Complexity, Prentice Hall, New Jersey, 1981.
[12] V.Th. Paschos, Complexité et approximation polynomiale, Hermès, Paris, 2004.
[13] H.R. Lewis, C.H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1981.
[14] R.E. Ladner, On the structure of polynomial time reducibility, J. Assoc. Comput. Mach. 22 (1975) 155–171.
[15] R.E. Ladner, N.A Lynch, A.L Selman, A comparison of polynomial time reducibilities, Theoret. Comput. Sci. 1 (1975) 103–123.
[16] G. Ausiello, A. D'Atri, M. Protasi, Structure preserving reductions among convex optimization problems, J. Comput. System Sci. 21 (1980) 136–153.
[17] M. Demange, V.Th. Paschos, On an approximation measure founded on the links between optimization and polynomial approximation theory, Theoret. Comput. Sci. 158 (1996) 117–141.
[18] A. Aiello, E. Burattini, M. Furnari, A Massarotti, F. Ventriglia, Computational complexity: The problem of approximation, in: J. Demetrovic, G. Katona, A. Salonaa (Eds.), Algebra, Combinatorics, and Logic in Computer Science, vol. I, in: Colloquia Mathematica Societatis János Bolyai, North-Holland, New York, 1986, pp. 51–62.
[19] E. Zemel, Measuring the quality of approximate solutions to zero-one programming problems, Math. Oper. Res. 6 (1981) 319–332.
[20] M. Demange, V.Th. Paschos, Quelques étapes vers la conciliation de la théorie d'approximation et celle d'optimisation: une nouvelle théorie d'approximation polynomiale et résultats préliminaires, C. R. Acad. Sci. Paris Sér. I Math. 317 (1993) 409–414.
[21] M. Demange, P. Grisoni, V.Th. Paschos, Differential approximation algorithms for some combinatorial optimization problems, Theoret. Comput. Sci. 209 (1998) 107–122.
[22] M. Demange, V.Th. Paschos, Asymptotic differential approximation ratio: Definitions, motivations and application to some combinatorial problems, RAIRO Oper. Res. 33 (1999) 481–507.
[23] J. Håstad, Clique is hard to approximate within $n^{1-\epsilon}$, Acta Mathematica 182 (1999) 105–142.
[24] C. Bazgan, V.Th. Paschos, Differential approximation for optimal satisfiability and related problems, European J. Oper. Res. 147 (2) (2003) 397–404.
[25] M. Szegedy, Hardness of approximation, in: T.F. Gonzalez (Ed.), Handbook of Approximation Algorithms and Metaheuristics, Chapman & Hall (chapter 17), Boca Raton, 2007, pp. 17–1–17–16.
[26] V.Th. Paschos, Polynomial approximation and graph coloring, Computing 70 (2003) 41–86.
[27] S. Sahni, T. Gonzalez, P-complete approximation problems, J. Assoc. Comput. Mach. 23 (1976) 555–565.
[28] L. Babai, L. Fortnow, L.A. Levin, M. Szegedy, Checking computations in polylogarithmic time, in: Proc. STOC'91, 1991, pp. 21–31.

[29] U. Feige, S. Goldwasser, L. Lovász, S. Safra, M. Szegedy, Interactive proofs and the hardness of approximating cliques, J. Assoc. Comput. Mach. 43 (2) (1996) 268–292.
[30] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and intractability of approximation problems, J. Assoc. Comput. Mach. 45 (3) (1998) 501–555.
[31] J. Håstad, Some optimal inapproximability results, J. Assoc. Comput. Mach. 48 (2001) 798–859.
[32] U. Feige, A threshold of $\ln n$ for approximating set cover, J. Assoc. Comput. Mach. 45 (1998) 634–652.
[33] M. Bellare, O. Goldreich, M. Sudan, Free bits and non-approximability — Towards tight results, SIAM J. Comput. 27 (3) (1998) 804–915.
[34] U. Feige, J. Kilian, Zero knowledge and the chromatic number, in: Proc. Conference on Computational Complexity, 1996, pp. 278–287.
[35] C. Lund, M. Yannakakis, The approximation of maximum subgraph problems, in: Proc. ICALP'93, in: Lecture Notes in Computer Science, vol. 700, Springer-Verlag, 1993, pp. 40–51.
[36] C. Lund, M. Yannakakis, On the hardness of approximating minimization problems, J. Assoc. Comput. Mach. 41 (5) (1994) 960–981.
[37] S. Arora, C. Lund, Hardness of approximation, in: D.S. Hochbaum (Ed.), Approximation Algorithms for NP-hard Problems, PWS, Boston, 1997, pp. 399–446 (chapter 10).
[38] I. Dinur, The PCP theorem by gap amplification, J. Assoc. Comput. Mach. 54 (3) (2007).
[39] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, in: R.M. Karp (Ed.), Complexity of Computations, American Mathematics Society, 1974, pp. 43–73.
[40] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation and complexity classes, J. Comput. System Sci. 43 (1991) 425–440.
[41] G. Ausiello, A. D'Atri, M. Protasi, Lattice-theoretical ordering properties for NP-complete optimization problems, Fund. Inform. 4 (1981) 83–94.
[42] A. Paz, S. Moran, Non deterministic polynomial optimization problems and their approximations, Theoret. Comput. Sci. 15 (1981) 251–277.
[43] P. Orponen, H. Mannila, On approximation preserving reductions: Complete problems and robust measures. Technical Report C-1987-28, Dept. of Computer Science, University of Helsinki, Finland, 1987.
[44] C. Berge, Graphs and Hypergraphs, North Holland, Amsterdam, 1973.
[45] M.W. Krentel, The complexity of optimization problems, J. Comput. System Sci. 36 (1988) 490–509.
[46] Ph.G. Kolaitis, M.N. Thakur, Approximation properties of NP minimization classes, J. Comput. System Sci. 50 (3) (1995) 391–411.
[47] P. Crescenzi, V. Kann, R. Silvestri, L. Trevisan, Structure in approximation classes, SIAM J. Comput. 28 (5) (1999) 1759–1782.
[48] P. Crescenzi, R. Silvestri, L. Trevisan, Natural complete and intermediate problem in approximation classes. Manuscript, 1994.
[49] P. Berman, G. Schnitger, On the complexity of approximating the independent set problem, Inform. Comput. 96 (1992) 77–94.
[50] V. Kann, Polynomially bounded problems that are hard to approximate, Nordic J. Computing 1 (1994) 317–331.
[51] P. Crescenzi, A. Panconesi, Completeness in approximation classes, Inform. Comput. 93 (2) (1991) 241–262.
[52] P. Crescenzi, L. Trevisan, On approximation scheme preserving reducibility and its applications, Theory Comput. Syst. 33 (1) (2000) 1–16.
[53] S. Khanna, R. Motwani, M. Sudan, U. Vazirani, On syntactic versus computational views of approximability, SIAM J. Comput. 28 (1998) 164–191.
[54] W. Fernandez de la Vega, G.S. Lueker, Bin packing can be solved within $1+\epsilon$ in linear time, Combinatorica 1 (4) (1981) 349–355.
[55] P. Crescenzi, L. Trevisan, Max NP-completeness made easy, Theoret. Comput. Sci. 225 (1–2) (1999) 65–79.
[56] C. Bazgan, B. Escoffier, V.Th. Paschos, Completeness in standard and differential approximation classes: Poly-(D)APX- and (D)PTAS-completeness, Theoret. Comput. Sci. 339 (2005) 272–292.
[57] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, J. Assoc. Comput. Mach. 41 (1) (1994) 153–180.
[58] V. Chvátal, A greedy-heuristic for the set covering problem, Math. Oper. Res. 4 (1979) 233–235.
[59] P. Slavík, A tight analysis of the greedy algorithm for set cover, in: Proc. STOC'96, 1996, pp. 435–441.
[60] R. Raz, S. Safra, A sub-constant error probability low-degree test and a sub-constant error probability PCP characterization of NP, in: Proc. STOC'97, 1997, pp. 475–484.
[61] B. Escoffier, V.Th. Paschos, Completeness in approximation classes beyond apx, Theoret. Comput. Sci. 359 (1-3) (2006) 369–377.
[62] S. Arora, M. Sudan, Improved low-degree testing and its applications, Combinatorica 23 (3) (2003) 365–426.
[63] R. Duh, M. Fürer, Approximation of k-set cover by semi-local optimization, in: Proc. STOC'97, 1997, pp. 256–265.
[64] R. Hassin, J. Monnot, The maximum saving partition problem, Oper. Res. Lett. 33 (2005) 242–248.
[65] J. Monnot, V.Th. Paschos, S. Toulouse, Approximation algorithms for the traveling salesman problem, Math. Methods Oper. Res. 57 (1) (2003) 387–405.
[66] J. Monnot, V.Th. Paschos, S. Toulouse, Differential approximation results for the traveling salesman problem with distances 1 and 2, European J. Oper. Res. 145 (3) (2002) 557–568.
[67] C. Bazgan, R. Hassin, J. Monnot, Approximation algorithms for some vehicle routing problems, Discrete Appl. Math. 146 (1) (2005) 27–42.
[68] R. Hassin, S. Khuller, z-approximations, J. Algorithms 41 (2001) 429–442.
[69] M. Demange, J. Monnot, V.Th. Paschos, Bridging gap between standard and differential polynomial approximation: The case of bin-packing, Appl. Math. Lett. 12 (1999) 127–133.
[70] M. Demange, J. Monnot, V.Th. Paschos, Maximizing the number of unused bins, Found. Comput. Decision Sci. 26 (2) (2001) 169–186.
[71] C. Bazgan, J. Monnot, V.Th. Paschos, F. Serrière, On the differential approximation of MIN SET COVER, Theoret. Comput. Sci. 332 (2005) 497–513.
[72] B. Escoffier, V.Th. Paschos, Differential approximation of MIN SAT, MAX SAT and related problems, European J. Oper. Res. 181 (2) (2007) 620–633.
[73] M. Demange, Polynomial approximation of NP-hard problems and linear programming: A new approximation measure and algorithms. Ph.D. Thesis, CERMSEM, Université Paris I, 1994 (in French).
[74] J. Monnot, V.Th. Paschos, S. Toulouse, Optima locaux garantis pour l'approximation différentielle, Technique et Science Informatiques 22 (3) (2003) 257–288.
[75] J. Monnot, Critical families of instances and polynomial approximation. Ph.D. Thesis, LAMSADE, University Paris-Dauphine, 1998 (in French).
[76] S. Toulouse, Polynomial approximation: Local optima and differential ratio. Ph.D. Thesis, LAMSADE, Université Paris-Dauphine, 2001, (in French).
[77] G. Ausiello, C. Bazgan, M. Demange, V.Th. Paschos, Completeness in differential approximation classes, Internat. J. Found. Comput. Sci. 16 (6) (2005) 1267–1295.

[78] H.U. Simon, On approximate solutions for combinatorial optimization problems, SIAM J. Disc. Math. 3 (2) (1990) 294–310.

[79] G. Ausiello, M. Protasi, NP optimization problems and local optima, in: Y. Alavi, A. Schwenk (Eds.), Graph Theory. Combinatorics and Applications. Proc. Quadriennal International Conference on the Theory and Applications of Graphs, volume 2, Wiley and Sons, 1995, pp. 957–975.

[80] B. Escoffier, V.Th. Paschos, Proving completeness by logic, International Journal of Computer Mathematics 82 (2) (2005) 151–161.

[81] Ph.G. Kolaitis, M.N. Thakur, Logical definability of NP-optimization problems, Inform. Comput. 115 (2) (1994) 321–353.

[82] L. Trevisan, Inapproximability of combinatorial optimization problems, The Computing Research Repository, 2004.